

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF TELECOMMUNICATIONS

PODPORA QOS V MANET SÍTÍCH

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

AUTOR PRÁCE  
AUTHOR

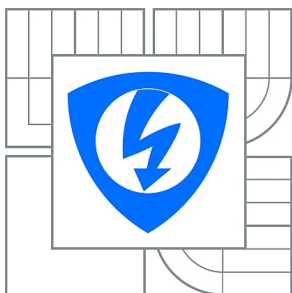
Bc. ROMAN FIGURNY

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF TELECOMMUNICATIONS

## PODPORA QOS V MANET SÍTÍCH

SUPPORT OF QOS IN MANET NETWORKS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. ROMAN FIGURNÝ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PAVEL VAJSAR

BRNO 2012



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav telekomunikací

# Diplomová práce

magisterský navazující studijní obor  
**Telekomunikační a informační technika**

**Student:** Bc. Roman Figurny

**ID:** 106423

**Ročník:** 2

**Akademický rok:** 2011/2012

**NÁZEV TÉMATU:**

## Podpora QoS v MANET sítích

### POKYNY PRO VYPRACOVÁNÍ:

V rámci diplomové práce bude nutné seznámit se s MANET (Mobile ad hoc) sítěmi a prostředím OPNET Modeler. Prostudovat a teoreticky zpracovat stávající možnosti řešení kvality služeb (QoS - Quality Of Services) u těchto sítí. V prostředí OPNET Modeler vytvořit model MANET sítě, který bude obsahovat podporu kvality služeb. Provést simulaci a analyzovat strukturu nastavení kvality služeb. Popsat dosažené výsledky simulace. V druhé části práce poté realizovat MANET síť s podporovou směrovacího protokolu (např. OLSR, AODV, DSR). V této síti analyzovat provoz některým z paketových analyzátorů a to zejména provoz daného směrovacího protokolu. Prostudovat implementaci směrovacího protokolu a navrhnout příslušné modifikace.

### DOPORUČENÁ LITERATURA:

[1] MOHAPATRA, Prasant; LI, Jian; GUI, Chao. QoS in mobile ad hoc networks. IEEE Wireless Communications [online]. 2003, 1, [cit. 2011-10-14]. Dostupný z WWW: <<http://www.ieee.org/index.html>>.

**Termín zadání:** 6.2.2012

**Termín odevzdání:** 24.5.2012

**Vedoucí práce:** Ing. Pavel Vajsar

**Konzultanti diplomové práce:**

**prof. Ing. Kamil Vrba, CSc.**

*Předseda oborové rady*

### UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Výzkum popsáný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

## **ANOTACE**

Teoretická část diplomové práce se zabývala problematikou MANET sítí, problematikou zajištění požadované kvality služeb v MANET sítích a rozbořem možných implementací kvality služeb do těchto sítí.

Praktická část práce byla zaměřena na popis upraveného procesního modelu OLSR s podporou kvality služeb. Dále byla v programu OPNET Modeler navržena topologie MANET sítě a byla provedena simulace nejprve bez podpory kvality služeb a poté byla provedena simulace s podporou kvality služeb. Výsledky simulací byly porovnány, rozebrány a graficky zpracovány. V další části práce byla navržena reálná MANET síť s podporou směrovacího protokolu OLSR. Pomocí programu Wireshark byla zachycena síťová komunikace daného směrovacího protokolu. V poslední části práce byl nejprve popsán význam a funkce zásuvných modulů obsažených v implementaci směrovacího protokolu OLSR a nakonec byl navržen nový zásuvný modul, který se stará a rozesílání informací o stavu linek mezi jednotlivými uzly MANET sítě.

## **KLÍČOVÁ SLOVA**

MANET, QoS, OPNET Modeler , OLSR, QOLSR, OLSR démon, Wireshark

## **ABSTRACT**

The theoretical part of Master's thesis has been dealing with issues of MANET networks, issues of providing quality of services in MANET networks and analysis of various implementations of quality of services in these networks.

At the start of the practical part of this thesis was described modified OLSR process model with support of quality of services. In simulation program OPNET Modeler was designed a topology of MANET network, firstly was run a simulation of this network without any support of quality of services and then was run a simulation with support of quality of services. The results of these simulations were analyzed, compared and graphically interpreted. In the next part of this thesis was designed a real MANET network with OLSR support. OLSR network communication was captured with the use of Wireshark. In the last part of this thesis was firstly described a meaning and functions of plugins in OLSR implementation and then was developed a plugin which is responsible for sending messages which contain a state of the links between nodes of MANET networks.

## **KEYWORDS**

MANET, QoS, OPNET Modeler , OLSR, QOLSR, OLSR daemon, Wireshark



Prohlašuji, že svoji diplomovou práci na téma : *Podpora QoS (Quality of Services) v MANET sítích* jsem vypracoval (-a) samostatně pod vedením vedoucího diplomové práce s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, v souvislosti s vytvořením této práce jsem neporušil (-a) autorská práva třetích osob, zejména jsem nezasáhl (-a) nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom (-a) následků porušení ustanovení §11 a následujících autorského zákona č.121/2000Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení §152 trestního zákona č.140/1961Sb.

V Brně dne .....

.....  
Podpis autora



# Poděkování

Děkuji vedoucímu práce Ing. Pavlu Vajsarovi za velmi užitečnou metodickou pomoc a cenné rady při zpracování diplomové práce.

V Brně dne .....

.....  
podpis autora

# OBSAH

<b>ÚVOD</b>	<b>12</b>
<b>1 BEZDRÁTOVÉ SÍŤE</b>	<b>13</b>
1.2 MANET síť.....	14
1.1.1 Dělení MANET sítí.....	15
1.1.2 Architektura MANET sítí.....	16
<b>2 KVALITA SLUŽEB V MANET SÍTÍCH</b>	<b>19</b>
2.1 Hlavní problémy při zajišťování kvality služeb.....	19
2.2 Implementace kvality služeb.....	22
2.3 Protokol OLSR - Optimized Link State Routing protocol.....	23
2.3.1 Využití protokolu OLSR.....	23
2.3.2 MPR – Multipoint Relaying.....	23
2.3.3 Zprávy využívané protokolem OLSR.....	24
2.3.4 Struktura OLSR paketu.....	25
2.4 Protokol QOLSR.....	26
<b>3 ZAVEDENÍ KVALITY SLUŽEB</b>	<b>27</b>
3.1 Out-of-band signalizace.....	27
3.2 In-band signalizace.....	27
3.3 Protokol INSIGNIA.....	28
<b>4 SIMULACE V PROGRAMU OPNET MODELER</b>	<b>32</b>
4.1 OPNET Modeler.....	32
4.1.1 Editory.....	32
4.2 Model OLSR s podporou QoS.....	34
<b>5 NÁVRH SIMULACE</b>	<b>36</b>
5.1 Návrh a popis modelové situace.....	36
5.1.1 Návrh scénáře.....	37
5.2 Zobrazení a vyhodnocení výsledků simulace.....	42
5.2.1 Koncové zpoždění.....	42
5.2.2 Kolísání zpoždění.....	44
5.2.3 Kvalita hovoru.....	45
<b>6 IMPLEMENTACE SMĚROVACÍCH PROTOKOLŮ</b>	<b>47</b>
6.1 Stav implementací směrovacích protokolů.....	47
6.2 Nasazení protokolu OLSR do reálné MANET sítě.....	49

<b>7</b>	<b>REALIZACE REÁLNÉ MANET SÍTĚ</b>	<b>50</b>
7.1	Základní návrh realizace.....	50
7.2	OLSR démon.....	51
<b>8</b>	<b>ANALÝZA SÍŤOVÉ KOMUNIKACE</b>	<b>61</b>
8.1	Analýza OLSR paketů.....	61
<b>9</b>	<b>ZÁSUVNÉ MODULY OLSR DÉMONA</b>	<b>65</b>
9.1	Princip zásuvných modulů.....	65
9.2	Návrh zásuvného modulu s podporou kvality služeb.....	67
9.2.1	Návrh OLSR zprávy s podporou kvality služeb.....	67
9.3	Rozbor zásuvného modulu s podporou kvality služeb.....	68
9.4	Instalace a konfigurace QoS zásuvného modulu.....	71
9.5	Analýza OLSR paketu s obsahem nové OLSR zprávy.....	74
	<b>ZÁVĚR</b>	<b>75</b>
	<b>LITERATURA</b>	<b>76</b>
	<b>SEZNAM ZKRATEK</b>	<b>79</b>

# ÚVOD

Mobilní bezdrátová komunikace prošla v posledních letech obrovským rozmachem. Mobilní zařízení, jako jsou například mobilní telefony, notebooky, tablety a další podobná mobilní zařízení, se vyskytují už téměř v každé domácnosti. Objevuje se také stále více uživatelů, kteří chtějí mít přístup k Internetu, bez ohledu na to, kde se právě nacházejí. Charakter zařízení, která budou mít přístup k informacím z jakéhokoli místa, umožňují právě bezdrátové technologie, které jsou také nejjednodušším nástrojem pro jejich vzájemné propojení.

Vývoj bezdrátových technologií se posouvá kupředu a objevují se nové komunikační modely a systémy, kde jsou samotní mobilní uživatelé součástí komunikační sítě a zajišťují její funkčnost. Takovéto systémy jsou označovány jako mobilní ad-hoc sítě (Mobile Ad-hoc Networks – dále jen MANET).

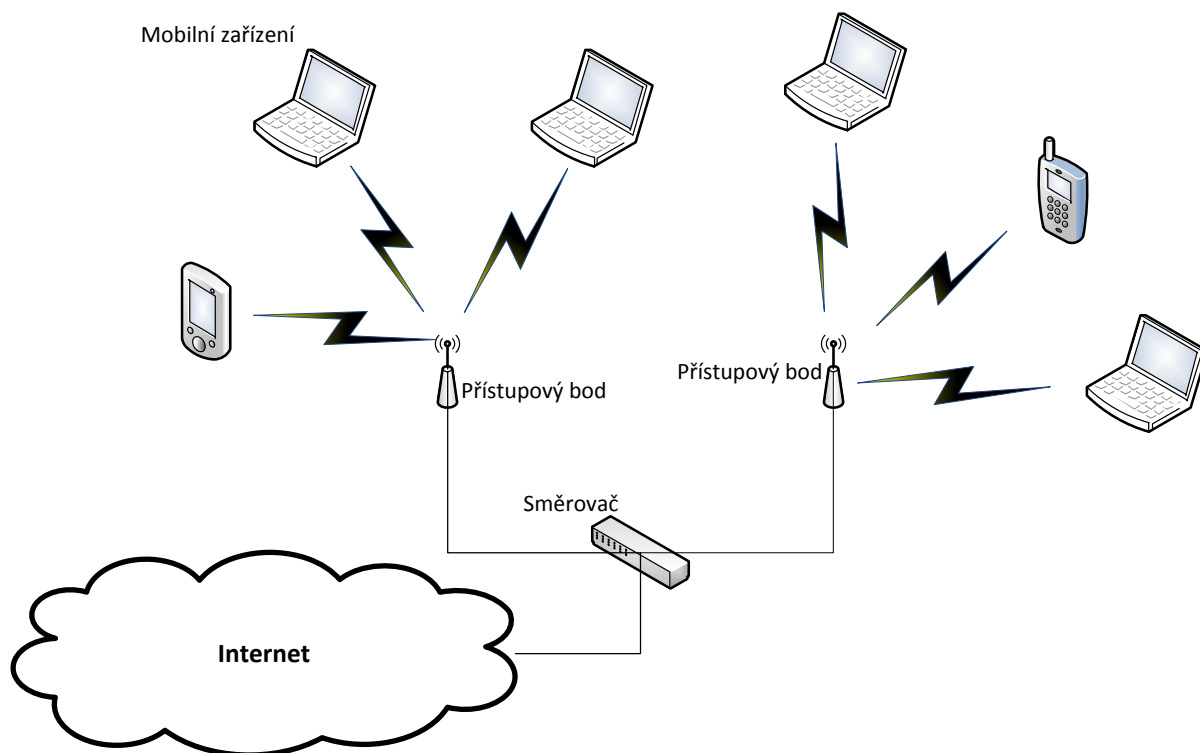
Mobilní ad-hoc síť je tvořena souborem bezdrátových uzlů, které mohou vzniknout kdekoli a v jakémkoli okamžiku bez použití jakékoli předchozí síťové infrastruktury. Jedná se o autonomní systém, kde uživatelé komunikují pomocí bezdrátové technologie a nejsou nijak omezeni svou polohou nebo její změnou, často také sami fungují jako směrovače. V tomto typu sítě je velmi obtížné vyřešit úlohu směrování a bezpečnosti. Mnohem obtížnější je pak ale v této síti zaručit určitou kvalitu služeb, která je při vzrůstající poptávce po multimediálním obsahu také velmi potřebná.

Teoretická část této diplomové práce bude zaměřena na objasnění základních pojmů kolem MANET sítí a implementace zajištění požadované kvality služeb do MANET sítí.

V praktické části této práce byla pomocí programu OPNET Modeler vytvořena topologie MANET sítě a poté byly provedeny simulace nejprve bez podpory kvality služeb a následně také s podporou kvality služeb. Další část práce byla zaměřena na nasazení směrovacího protokolu OLSR do reálné MANET sítě. Byly rozebrány zásuvné moduly implementace protokolu OLSR a následně byl navržen nový zásuvný modul starající se o zasílání informací o stavu linek mezi uzly MANET sítě.

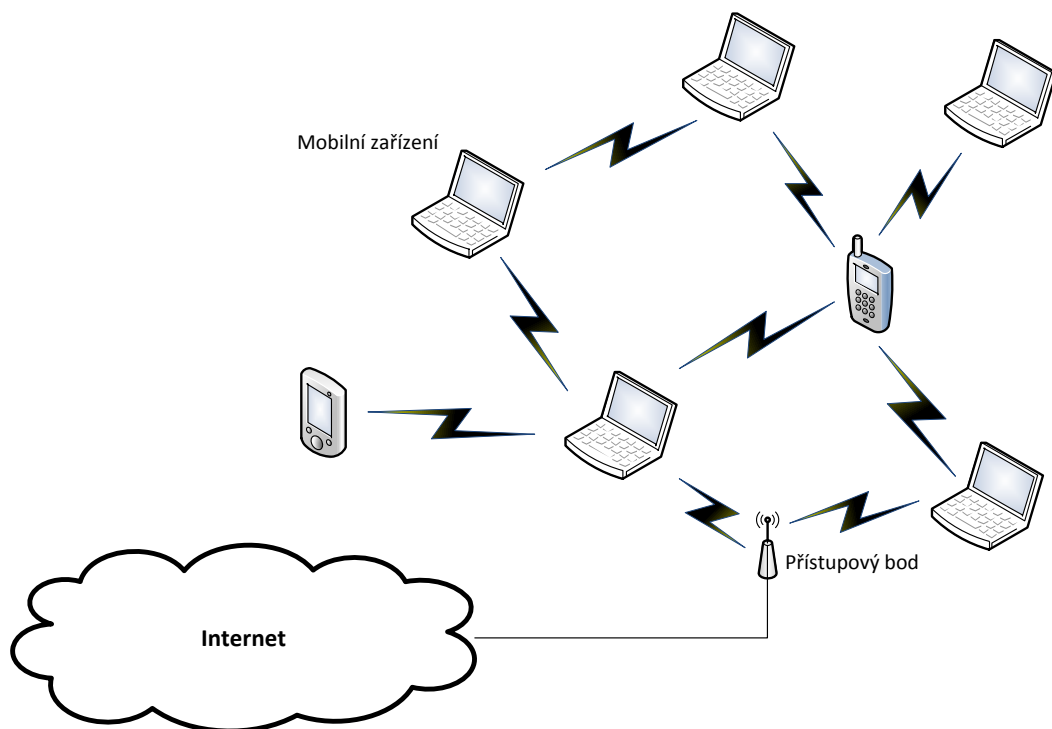
# 1 BEZDRÁTOVÉ SÍTĚ

Mobilní bezdrátové sítě dělíme podle typu infrastruktury na dva základní typy. Prvním typem je bezdrátová síť připojena na přístupový bod, jednotlivá bezdrátová zařízení spolu mohou komunikovat jen pomocí přístupového bodu. Typické schéma je znázorněno na obrázku 1.1



Obrázek č. 1.1: Bezdrátová síť s pevnou infrastrukturou

Druhý typ bezdrátové sítě je označován jako Mobile Ad-hoc Network neboli MANET síť. Jedná se dynamickou síť, jejíž uzly mezi sebou dokážou komunikovat i bez přítomnosti pevného přístupového bodu. Jednotlivá bezdrátová zařízení mohou zastupovat funkci směrovače. Základní princip je znázorněn na obrázku 1.2



Obrázek č. 1.2 : Bezdrátová síť typu Ad-hoc

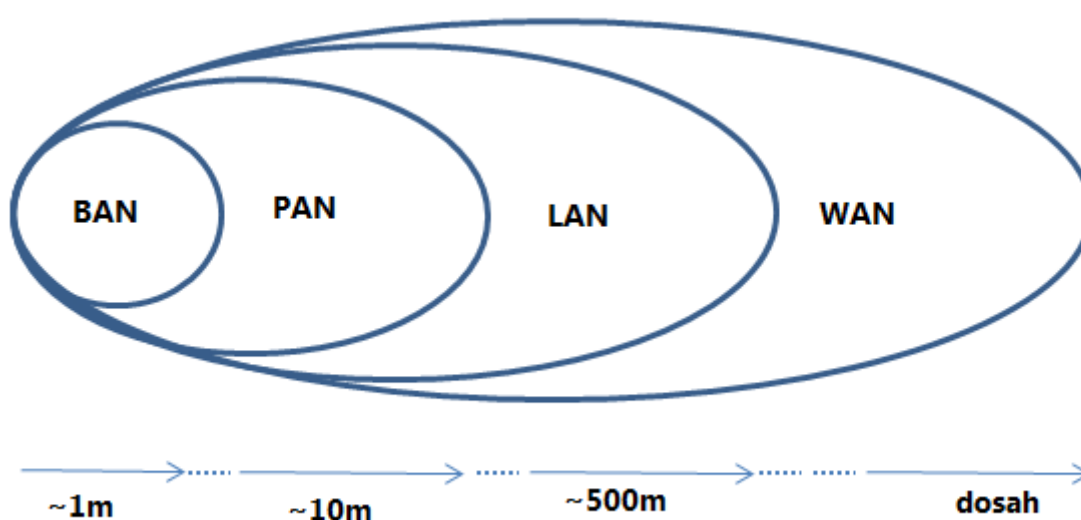
## 1.2 MANET síť

V počátcích 70 let 19. století byly MANET sítě známy pod názvem Packet Radio Networks, neboli paketové radiové sítě. Tyto sítě byly sponzorovány společností DARPA (Defence Advanced Research Project Agency), která se zabývala projektem, který obsahoval několik bezdrátových terminálů, které byly schopné spolu komunikovat mezi sebou na bojišti. Toto dalo základ pro další vývoj těchto sítí. MANET síť si tedy můžeme představit jako dynamickou bezdrátovou síť, která se vytváří ze skupiny mobilních uzlů, které mezi sebou komunikují na sdíleném bezdrátovém komunikačním kanálu. Jako mobilní uzly označujeme všechny bezdrátové komunikační zařízení, jako jsou notebooky, mobilní telefony, tablety nebo jakákoli jiná zařízení s podporou bezdrátové technologie. Mobilní uzly mohou vznikat kdekoli a v jakémkoli okamžiku bez použití jakékoli předem existující síťové infrastruktury. Poloha těchto mobilních uzlů je náhodná a mohou se tedy vyskytovat například v letadlech, lodích a automobilech. Jedná se o autonomní systém, kde mobilní uzly jsou propojené pomocí bezdrátové technologie a mohou se neomezeně pohybovat, nemusí mít tedy fixní polohu. Mobilní Ad-hoc sítě mohou být využívány v mnoha oblastech, samozřejmě první oblast nasazení těchto sítí bylo prostředí armádní, mezi další oblast použití

můžeme zmínit použití v případě nouzových událostí, dále mohou být tyto sítě použity v oblasti bezdrátových senzorů a samozřejmě mohou mít také architekturu mesh sítě.

### 1.1.1 Dělení MANET sítí

MANET sítě můžeme rozdělit například podle rozlohy jejich pokrytí. Existují čtyři hlavní třídy, Body Area Network (BAN), Personal Area Network (PAN), Local Area Network (LAN) a Wide Area Network (WAN). Přibližná rozloha pokrytí jednotlivých tříd je zobrazena na obrázku č. 1.3.[10]



Obrázek č. 1.3: Pokrytí jednotlivých tříd MANET sítí

#### a) BAN - Body Area Network

Do této třídy bychom mohli zařadit například mikrofon, sluchátka a další podobné přístroje, které můžeme nosit na těle. A právě tato technologie by měla poskytovat spojení mezi těmito zařízeními. Komunikační dosah by měl být v rozmezí velikosti lidského těla, to znamená poloměr 1-2 metry. Mezi základní požadavky na tuto třídu patří schopnost propojit zařízení v rozsahu úplných zařízení (mobilní telefon) k jejich částem (mikrofon, sluchátka), dále schopnost autokonfigurace (přidání nebo odebrání zařízení z BAN by mělo být transparentní uživateli), integrace služeb (přenos multimediálních dat musí existovat společně s obyčejnými daty, například daty z Internetu) a schopnost komunikovat s ostatními sítěmi typu BAN (výměna dat s ostatními uživateli) nebo sítěmi typu PAN (přístup k Internetu).

#### **b) PAN - Personal Area Network**

Na rozdíl od BAN sítí, která propojuje zařízení, které uživatel nosí na sobě, PAN síť se snaží o propojení zařízení kolem uživatele v rozsahu až 10 metrů. PAN síť tedy umožňuje propojení BAN sítí vyskytujících se ve vzájemné blízkosti a také poskytují možnost propojení BAN sítí, které nejsou ve vzájemné blízkosti.

#### **c) Wireless Local Area Network (WLAN)**

Hlavní výhoda WLAN sítí oproti LAN sítím je ta, že nabízejí větší flexibilitu. A právě tato vlastnost je hlavním tahounem nástupu WLAN sítí v domácnostech a kancelářích. Typický rozsah těchto sítí je 100 až 500 metrů. WLAN síť samozřejmě musí splňovat požadavky jako LAN síť, mezi něž patří například plná konektivita mezi stanicemi, schopnost všesměrového vysílání nebo vysoká kapacita linky. Splnit tyto požadavky není jednoduché, jelikož WLAN síť musí čelit určitým problémům typickým pro bezdrátové prostředí, jako jsou například bezpečnost přenosu, spotřeba energie, mobilita nebo šířka pásma limitována bezdrátovým prostředím.

#### **d) WAN – Wide Area Network**

Jedná se o stejný koncept jako u LAN sítí, jen rozsah pokrytí je větší (nad 500 metrů), viz obrázek č. 1.3.

#### **1.1.2 Architektura MANET sítí**

Uzly v MANET sítích můžeme rozdělit podle jejich funkcí v síti na dva typy. První z nich je klient neboli Small Mobile Host (SMH), který má jen zredukované možnosti úschovy dat, komunikace a zdroje energie. Druhý z těchto typů je označován jako server neboli Large Mobile Host (LMH), který má obvykle větší výpočetní výkon a lepší možnosti úschovy dat než SMH. Jelikož LMH má větší rozsah zdrojů než SMH, tak LMH uzly obsahují kompletní DBMS (Database Management System) databázi a přebírají hlavní zodpovědnost za všesměrové vysílání a požadavky klientů. [10]

Klienti mají obvykle jen dostatečné množství zdrojů na to, aby uschovaly alespoň částečnou DBMS databázi. Každý uzel pokrývá určitou oblast své působnosti. U LMH je samozřejmě tato oblast větší než u SMH, jelikož u LMH se předpokládá, že bude



mít dostatečný zdroj energie. Samozřejmě se snižující se hladinou energie dochází k ubývání pokrytí oblastí jednotlivými uzly, a proto byly navrženy tři základní stavy, ve kterých se uzel může nacházet, aby docházelo k menší spotřebě energie:

- **Active Mode** (aktivní mód): dochází k největší spotřebě energie, uzel umožňuje vysílání i příjem dat.
- **Recieve Mode** (mód příjmu): procesor je schopen zpracovávat informace a je také schopen přijímat notifikace o zprávách od ostatních uzlů
- **Sleep Mode** (mód spánku): procesor nezpracovává informace a uzel nemá žádnou schopnost vysílat nebo přijímat zprávy, uzel je neaktivní. Tento stav umožňuje uzlu se vypnout na určitý časový interval bez toho, aniž by se musel znovu spouštět nebo inicializovat.

#### **Architektury pro poskytnutí konektivity v MANET sítích:**

##### **a) Hierarchická síťová architektura**

Tato architektura rozděluje celou síť na jednotlivé podsítě. Každá podsít' pak dynamicky určí uzel, který bude sloužit jako brána do ostatních podsítí. Toto buduje hierarchii mezi uzly. Mezi výhody této architektury patří především jednodušší správa sítě.

##### **b) Flat-routed architektura**

Neobsahuje žádný koncept podsítí a speciálních bran, každý uzel má stejnou zodpovědnost jako ostatní. Mezi výhody této architektury patří především větší spolehlivost, optimální směrování, omezení spotřeby bezdrátových zdrojů a lepší rozložení zátěže na jednotlivé uzly.

## **Typy komunikace v MANET sítích:**

### **a) Peer to Peer (rovný s rovným)**

Komunikace mezi dvěma uzly probíhá přímo, tedy bez potřeby přítomnosti mezilehlých uzlů.

### **b) Remote to Remote (vzdálený se vzdáleným)**

Komunikace mezi těmito uzly probíhá přes více než jeden skok a je zajištěno stabilní směrování mezi těmito uzly. To znamená, že je zajištěna stabilní cesta mezi těmito dvěma uzly. Na cestě se vyskytují mezilehlé uzly, které poskytují propojení. Tento případ může být výsledek toho, že několik uzlů zůstává v komunikačním dosahu mezi sebou v jedné oblasti, nebo se tyto uzly pohybují jako skupina.

### **c) Dynamic traffic (dynamický provoz)**

Tento typ provozu nastává v případě, když se jednotlivé uzly sítě náhodně připojují nebo odpojují, nebo pokud uzly neustále mění svou pozici. Cesty musí být při každé změně znovu sestaveny, toto vede ke špatné konektivitě a k síťově aktivitě vysílané v malých dávkách.

## **2 KVALITA SLUŽEB V MANET SÍTÍCH**

Bezdrátové mobilní ad-hoc sítě jsou složeny z mobilních uzlů, které mezi sebou komunikují pomocí určité bezdrátové technologie. Pro spojení dvou koncových uzlů může existovat přímá cesta, ale může také docházet při hledání koncového uzlu k několika přeskokům přes mezilehlé uzly sítě. MANET sítě nemají žádnou pevnou infrastrukturu nebo podporu centrální administrativy. Topologie takovýchto sítí se dynamicky mění, mobilní uzly dynamicky vznikají a zanikají, linky mezi uzly mohou přestat být stabilní, a proto je zajištění podpory kvality služeb u těchto sítí velmi obtížnou úlohou.

### **2.1 Hlavní problémy při zajišťování kvality služeb**

#### **a) Nespolehlivý komunikační kanál**

Hlavním nedostatkem bezdrátových sítí je nespolehlivý komunikační kanál, na kterém vznikají bitové chyby. Bitové chyby vznikají v důsledku výskytu interferencí, tepelného šumu atd. Tyto jevy vedou ke sníženému počtu doručených paketů. Dalším problémem komunikačního kanálu je ten, že signál se šíří vzduchem, a toto může vést k jeho jednoduchému odposlouchávání. [22]

#### **b) Složitá údržba komunikační cesty**

Údržba komunikačního kanálu mezi jednotlivými uzly sítě je velmi složitá, jelikož z principu MANET sítí vyplývá, že topologie sítě je proměnlivá a velmi nestabilní. Vytvořené směrovací cesty mohou přestat existovat dokonce i při procesu přenášení dat mezi uzly. Je tedy nutné, aby síť byla schopná tento problém okamžitě řešit a založit novou cestu jako náhradu té původní a to v co nejkratším čase a s minimální režií.

### **c) Postrádání centralizované kontroly**

Členové ad-hoc sítě se mohou připojit nebo odpojit v jakémkoli okamžiku, síť je vytvářena spontánně. V ad-hoc síti není zavedena žádná centrální kontrola uzlů sítě, a proto je pro zavedení kvality služeb potřeba větší komplexita a adaptace algoritmu, který se o zajištění služeb stará.

### **d) Mobilita uzlů**

Podstatou MANET sítě je výskyt uzlů, které se neustále a nepředvídatelně pohybují s různou rychlostí. Topologie těchto sítí proto musí být aktualizována velmi často, aby bylo možné zajistit doručení paketů ke všem požadovaným uzlům sítě, toto vede opět ke snížení počtu doručených paketů v určitém časovém intervalu.

### **e) Omezená zásoba energie**

Uzly MANET sítě jsou považovány za mobilní zařízení, které mají k dispozici omezené množství elektrické energie. Při zajišťování požadované kvality služeb dochází ke zvýšené spotřebě energie, a proto je nutno hledat řešení energeticky úsporná.

### **f) Bezpečnost**

Zajištění bezpečnosti může být také bráno jako jeden z parametrů zajištění kvality služeb. Zajištění nedostatečné bezpečnosti může vést ke zneužití přístupu na komunikační kanál a tím přerušit zajištěné parametry kvality služeb.

### **g) Omezená šířka pásma a její odhad**

U bezdrátových sítí je samozřejmě omezená šířka pásma mnohem větší problém než u sítí drátových. U mobilních uzlů MANET sítě je velmi těžké odhadnout šířku pásma, která je právě k dispozici, jelikož každý uzel má v různých okamžicích také různou dostupnou šířku pásma.

### **h) Různé aplikační nároky pro parametry kvality služeb**

Požadavky na parametry kvality služeb se také velmi liší u různých MANET sítí. Proto jeden standard pro zajištění kvality služeb pro různé aplikace v MANET sítích nebude dostatečný.

### **i) Heterogenita**

Sítě typu MANET jsou obvykle heterogenní s velkou škálou různých typů mobilních uzlů, pracujícími s různými technologiemi. Uzly mají různé výpočetní schopnosti a také jejich zásoba energie je různá, a proto budou mít různé uzly sítě různou zodpovědnost, síťovou aktivitu a spotřebu energie. Jak se vyrovnat s touto heterogenitou je klíčový faktor k úspěšnému fungování MANET sítí.

### **j) Problém skrytého uzlu**

Problém skrytého uzlu nastává tehdy, když dva signály, například uzlu (A) a uzlu (B), které jsou navzájem mimo vysílací dosah, dorazí zároveň do uzlu (C). V tomto případě dochází ke kolizi.

## 2.2 Implementace kvality služeb

Zajištění požadované kvality služeb v MANET sítích je realizováno pomocí směrovacích a signalizačních protokolů. Mezi signalizační protokoly řadíme například protokol INSIGNIA, který byl navržen výhradně pro MANET sítě. Směrovacích protokolů, které zajišťují požadovanou kvalitu služeb je velké množství, a můžeme je třídit podle různých kritérií. [20]

### Rozdělení směrovacích protokolů podle způsobu hledání směrovacích cest:

#### a) Proaktivní protokoly:

Směrovací tabulka je udržována na každém uzlu sítě. Tyto tabulky jsou pravidelně aktualizovány z každého ke každému uzlu, to znamená, že uzel může získat cestu okamžitě bez čekání. Výhodou těchto protokolů je především malá latence, nevýhodou zase větší výpočetní nároky na uzly.

Příklady: QOLSR, PLBQR

#### b) Reaktivní protokoly:

Jedná se o protokoly tzv. on-demand (na vyžádání). To znamená, že nejsou na uzlech udržovány žádné záznamy o cestách, pokud na nich zrovna není zaznamenán síťový provoz. Výhodou těchto protokolů jsou například menší nároky na uzly, mezi nevýhody poté větší latence.

Příklady: QAODV, AQOR, TBR [21]

#### c) Hybridní protokoly:

Kombinují základní podstatu proaktivních a reaktivních protokolů.

Příklady: ZRP, CEDAR, QMRPD

Směrovací protokoly dále můžeme třídit podle dalších kritérií, například podle typu síťové infrastruktury, typu zaručovaných služeb atd.

## **2.3 Protokol OLSR - Optimized Link State Routing protocol**

Jedná se o směrovací protokol, který je využíván v sítích typu MANET. Protokol OLSR je řazen mezi proaktivní směrovací protokoly, to znamená, že uzly dané sítě si mezi sebou pravidelně vyměňují informace o topologii sítě. Z toho vyplývá, že je zmapována celá síť a je známa její kompletní topologie. Ke zjištění topologie sítě a ke zjištění stavů linek sítě jsou využívány dva typy zpráv a to zpráva HELLO a zpráva TC (Topology Control). Pravidelná výměna informací sice zvyšuje síťový provoz, ale zvýší se tím i pravděpodobnost správného doručení paketu. Každý uzel si pro zasílání těchto paketů vybírá uzly z vlastního okolí, tento princip významně snižuje redundanci vysílání zpráv při výběru trasy. Díky tomuto principu funguje protokol OLSR i na kanále, kde se vyskytuje mnoho chyb. [1][5]

### **2.3.1 Využití protokolu OLSR**

Protokol OLSR je vhodný především pro velké mobilní sítě, využívá tzv. hop-by-hop směrování, to znamená, že každý uzel používá své lokální informace pro směrování paketů. Protokol OLSR je vhodný především pro sítě, kde je datový provoz náhodný mezi více uzly. Dále se hodí v případě, kdy se komunikující páry mění, jelikož se zde žádná další kontrola provozu negeneruje.

### **2.3.2 MPR – Multipoint Relaying**

Účel MPR je takový, že slouží pro snížení počtu vysílaných zpráv, které by byly v některých oblastech sítě nadbytečné. Toto snížení počtu zasílaných zpráv vede k minimalizaci energetických nároků na uzly a také snižuje zatížení šířky pásma pro jednotlivé linky. Každý uzel vybírá ze svých sousedů svého MPR, který je od daného uzlu vzdálený pouze jeden skok a musí umožnit spojení na vzdálenost dvou skoků. Pokud nastane situace, že existuje více sousedů umožňujících spojení s uzly na vzdálenost dvou skoků, pak MPR se stane ten uzel, který umožňuje spojení s největším počtem uzlů na dva skoky. V síti mohou přenášet všesměrové zprávy pouze MPR uzly. Všechny ostatní uzly mohou jen přijímat a zpracovat všesměrovou zprávu ale nesmí ji předat dalším uzlům v síti.

### **2.3.3 Zprávy využívané protokolem OLSR**

#### **Hello zpráva**

Zpráva Hello umožňuje zasílat informace o lokálních spojích a sousedech daného uzlu. Zprávy typu Hello jsou zasílány v pravidelných intervalech. Slouží k detekování sousedních uzlů a také slouží k signalizaci výběrového řízení MPR.

#### **TC (Topology Control) zpráva**

Pomocí zprávy TC získává uzel informace o stavech jeho sousedů a o stavech jednotlivých linek. Na tomto základě jsou informace o topologii sítě rozeslány do celé sítě. Uzly si mezi sebou zprávy typu TC zasílají v určitých pravidelných časových intervalech, ale pokud dojde k nějaké změně topologie v síti, dojde také k okamžitému rozeslání TC zpráv mezi uzly.

#### **MID (Multiple Interface Declaration) zpráva**

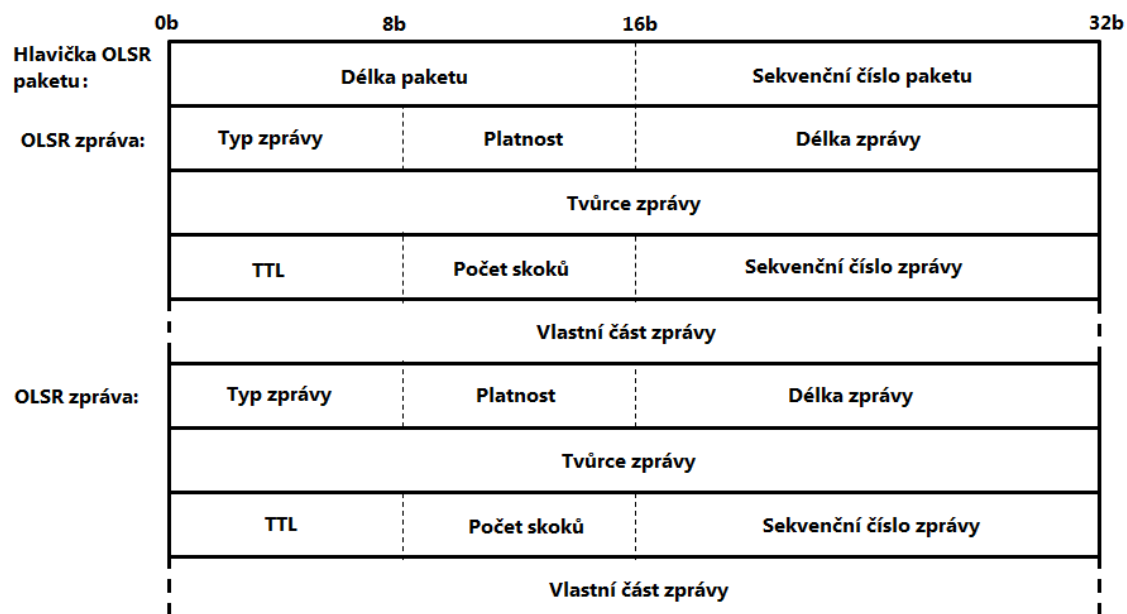
Jedná se o zprávu, která rozšiřuje informace o takových uzlech, které disponují v OLSR síti více rozhraními. Obsahují všechny IP adresy, které uzly používají na svých rozhraních.

#### **HNA (Host and Network Association) zpráva**

MANET síť založenou na OLSR protokolu je možno propojit s ostatními sítěmi, například ethernetovým spojením, pomocí něhož může být MANET síť napojena do sítě Internet. K tomuto účelu slouží právě zprávy typu HNA.



### 2.3.4 Struktura OLSR paketu



Obrázek č. 2.2: Struktura OLSR paketu

**Délka paketu** - délka celého paketu včetně záhlaví udávaná v bajtech

**Sekvenční číslo paketu** - zvyšuje se pokaždé, když je vytvořen nový OLSR paket, počáteční hodnota je generována náhodně

**Typ zprávy** - identifikuje typ zprávy, 0 - 127 rezervováno OLSR , 128 - 255 privátní využití

**Platnost** - platnost dané zprávy, uváděna v sekundách

**Délka zprávy** - délka dané OLSR zprávy uváděna v bajtech včetně záhlaví zprávy

**Tvůrce zprávy** – IP adresa tvůrce dané OLSR zprávy

**TTL (Time To Live)** – hodnota udává maximum skoků, při kterých může být zpráva zasílána dále v síti

**Počet skoků** – udává, kolikrát byla daná zpráva přeposlána dále v síti

**Vlastní část zprávy** - zde je možno nadefinovat vlastní část zprávy

## 2.4 Protokol QOLSR

Jedná se implementaci podpory QoS (kvality služeb) do směrovacího protokolu OLSR. Protokol QOLSR je proaktivní směrovací protokol s podporou kvality služeb pro MANET sítě. Pro zajištění kvality služeb jsou do původního protokolu OLSR implementovány následující funkce: [4], [15]

- a) Byla doplněna nová dodatečná pole do zpráv HELLO a TC, které nesou informace o jednotlivých parametrech sítě, jako jsou šířka pásma a zpoždění.
- b) Jsou měřeny metriky na všech spojkách.
- c) Na základě výsledků z těchto měření jsou vypočítány QoS-MPR (QMPR), výsledky v TC zprávách jsou publikovány v celé síti.

Protokol QOLSR využívá jako metriky především šířku pásma, zpoždění, jitter, ztrátovost a ochotnost. Informace o těchto metrikách jsou doplněny do jednotlivých zpráv HELLO a TC. Pomocí těchto informací jsou pak vypočítány QoS – MPR uzly a také jsou vypočítány směrovací tabulky v uzlech. QMPR je vybírán uzly sítě a slouží k šíření všesměrových zpráv v síti. QMPR uzly jsou vybírány podle určitých metrik, mezi defaultní metriky, při kterých se přihlíží k výběru QMPR uzlu jsou šířka pásma a zpoždění. Využívání tohoto typu uzlu redukuje provoz vysílání směrovacích zpráv a opakovaných vysílání.[4]

### **3 ZAVEDENÍ KVALITY SLUŽEB**

Zavedení požadované kvality služeb je na síťové vrstvě řešeno pomocí signalizačních protokolů, které zajišťují vytvoření, správu a ukončení datových toků mezi uzly sítě. Jelikož chování MANET sítí je velmi nevyzpytatelné a síť se velmi často mění, je třeba, aby tyto protokoly byly schopny na tyto situace velmi rychle reagovat. Existují dva základní typy signalizace a to tzv. out-of-band signalizace a tzv. in-band signalizace.

#### **3.1 Out-of-band signalizace**

Tento druh signalizace se vyznačuje tím, že informace zasílané pro zajištění zdrojů na jednotlivých uzlech jsou obsaženy ve speciálních zprávách. Mezi typické představitele této signalizace patří protokol RSVP (Resource Reservation Protocol). Tento typ signalizace je vhodný především pro pevné sítě. Out-of-band signalizace s sebou přináší velkou režii dat, neschopnost rychle reagovat na náhle změny v síti a pomalou rezervaci požadovaných zdrojů, proto tento typ signalizace není vhodný pro MANET sítě. [12]

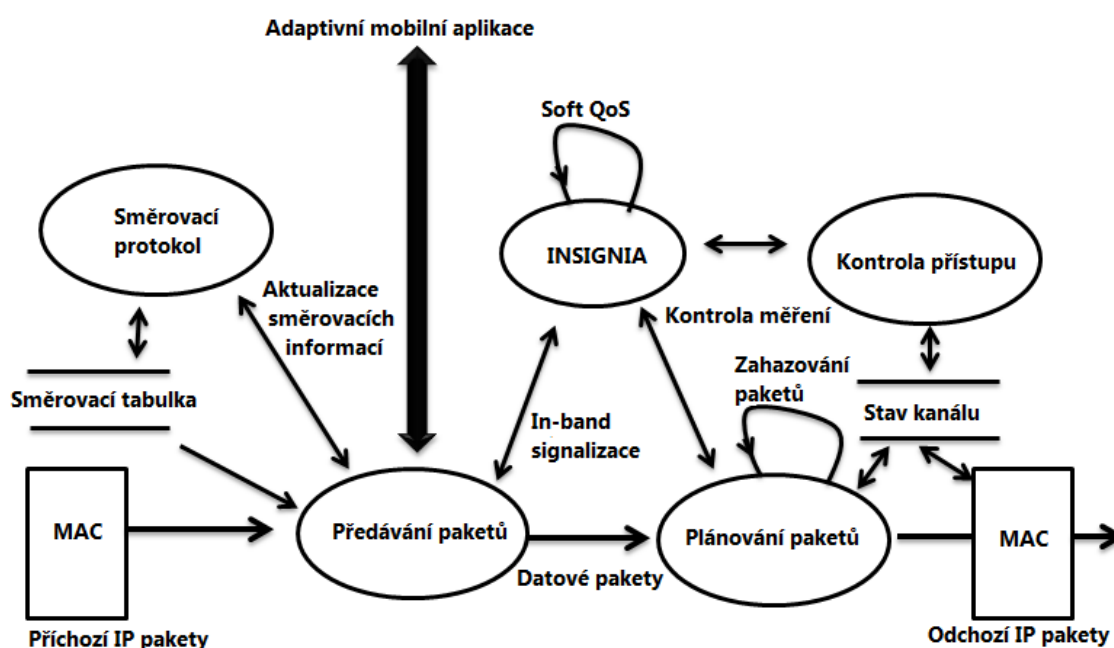
#### **3.2 In-band signalizace**

In-band signalizace se vyznačuje tím, že informace potřebné k rezervaci prostředků na jednotlivých uzlech sítě jsou obsaženy přímo v datovém IP paketu, není tedy potřeba zasílat žádné další speciální zprávy. Implementace těchto informací přímo do hlavičky IP paketů má za následek téměř okamžitou reakci sítě na vytvoření, správu nebo ukončení rezervačních prostředků pro datové toky. Tento typ signalizace je vhodný pro MANET sítě. Mezi tento typ signalizačních protokolů řadíme například protokol INSIGNIA nebo jeho upravenou verzi NR-INSIGNIA. [12]

### 3.3 Protokol INSIGNIA

Jedná se o první in-band signalizační protokol navržen výhradně pro potřeby MANET sítí. Protokol INSIGNIA je navržen pro podporu doručení dat služeb pracujících v reálném čase a vyznačuje se rychlým vytvořením, obnovou a správou komunikačních kanálů. INSIGNIA také hraje hlavní roli při zajišťování prostředků na mezilehlých uzlech pro komunikaci mezi zdrojovým a cílovým uzlem. [20]

Pro zaručení adaptivní kvality služeb pro aplikace pracujících v reálném čase v MANET sítích je třeba spravovat stavy datových toků, to znamená stav rezervace datových toků na jednotlivých uzlech. Adaptivní model služeb umožňuje specifikovat minimální a maximální nutnou šířku pásma pro pakety aplikací pracujících v reálném čase. Hlavní moduly správy datových toků jsou znázorněny na obrázku č. 3.1.



Obrázek č. 3.1: Moduly správy datových toků

**Modul předávání paketů** – klasifikuje příchozí pakety a předává je do příslušného modulu.

**Modul směrování** – dynamický sleduje změny v topologii ad-hoc sítí tím, že poskytuje směrovací informace všem mezilehlým uzlům. Správa datových toků předpokládá dostupnost MANET směrovacích protokolů, jako jsou například protokoly TORA,OLSR, DSR nebo AODV.

**Modul INSIGNIA** – stará se o založení, obnovu, adaptaci a ukončení datových toků aplikací pracujících v reálném čase.

**Modul kontroly přístupu** – přiřazuje šířku pásma jednotlivým datovým tokům podle informací obsažených v hlavičce IP paketu, tedy maximální a minimální potřebnou šířku pásma. Poté co jsou zdroje rezervovány, jsou v určitých časových intervalech obnovovány přijetím dalších IP paketů s informacemi o rezervaci potřebných zdrojů.

**Modul plánování paketů** – reaguje na podmínky v bezdrátovém kanále. To znamená, že odesílání paketů plánuje tak, aby nenastala situace, že některé uzly obdrží méně provozu než by mohly a některé uzly více než je potřeba.

**Modul kontroly přístupu na médium** – poskytuje požadovanou kvalitu služeb řízení přístupu ke sdílenému bezdrátovému médiu pro adaptivní služby pracující v reálném čase a pro služby typu best-effort.

In-band signalizace se vyznačuje tím, že informace potřebné k rezervaci prostředků jsou přenášeny přímo v hlavičce datových IP paketů. Protokol INSIGNIA využívá pole **IP volby** v hlavičce IP paketu. Jednotlivé položky v poli **IP volby** jsou zobrazeny na obrázku č. 3.2.

Mód rezervace	Typ služby	Indikátor zátěže	Indikátor šířky pásma	Požadovaná šířka pásma	
REQ/RES	RT/BE	BL/EL	MAX/MIN	MAX	MIN

Obrázek č. 3.2: Pole IP volby

**Mód rezervace** – pokud chce zdrojový uzel vytvořit datový tok k cílovému uzlu, nastaví se položka **mód rezervace** na stav **REQ** (požadavek). Po přijetí takovéhoho paketu reagují mezilehlé uzly tím, že modul kontroly přístupu na jednotlivých uzlech tento požadavek buď povolí, nebo ho zakáže. Pokud je požadavek zamítnut, tak jsou data doučeny tzv. best-effort. Pokud je stav v položce **mód rezervace** nastaven na hodnotu **RES** (rezervace), pak tento stav indikuje, že už byl požadavek na alokaci šířky pásma zaslán a byl povolen.

**Typ služby** – indikuje úroveň požadovaných služeb. Pokud je v položce **mód rezervace** nastavena hodnota **REQ** a v poli typ služby je nastavena hodnota **RT** (služba pracující v reálném čase), pak toto indikuje, že se jedná o pokus vytvořit datový tok aplikace pracující v reálném čase. Pokud jsou hodnoty nastaveny na hodnoty **RES/RT**, tak se jedná o stav, kdy požadovaný datový tok byl již sestaven. Datový tok může degradovat z hodnoty **RT** na hodnotu **BE** (best-effort) pokud dojde k situaci, že stará cesta zanikne a nově sestavená cesta není schopna zajistit požadované nároky na stávající datový tok.

**Požadovaná šířka pásma** – Toto pole umožňuje specifikovat maximum (**MAX**) a minimum (**MIN**) šířky pásma pro požadovaný datový tok. Hodnota **MIN** specifikuje doručení zátěže typu **základní úroveň**. Hodnota **MAX** indikuje podporu zátěže typu **základní úroveň plus rozšířená úroveň**.

**Indikátor zátěže** – Protokol INSIGNIA podporuje dva typy zátěže. První z nich je základní úroveň (**BL**), druhá z nich je rozšířená úroveň (**EL**). Rozšířená úroveň zátěže může být podporována, jen pokud jsou na všech mezilehlých uzlech dostatečné zdroje pro zajištění požadované šířky pásma. Tento stav se pozná tak, že na cílový uzel dorazí paket, který má v položce **mód rezervace** nastavenou hodnotu **REQ**, a který má v položce **indikátor zátěže** nastavenou hodnotu **MAX**. Pokud nejsou k dispozici dostatečné zdroje pro zajištění doručení rozšířené zátěže, pak je tato rozšířená zátěž doručována tzv. best-effort, tedy bez podpory kvality služeb.

**Indikátor šířky pásma** – Indikuje dostupnost zdrojů na mezilehlých uzlech mezi zdrojovým a cílovým uzlem. Pokud je indikátor nastaven na hodnotu **MAX**, pak jsou požadované zdroje na maximální šířku pásma splněny na všech mezilehlých uzlech, a je tedy podporována zátěž typu základní plus rozšířená úroveň. Pokud je položka nastavena na hodnotu **MIN**, tak minimálně jeden mezilehlý uzel není schopen zajistit maximální požadovanou šířku pásma pro datový tok. Indikátor **MIN** ale zajišťuje aspoň minimální požadovanou šířku pásma pro datový tok a tak je aspoň zátěž typu základní úroveň doručena s požadovanou kvalitou služeb.

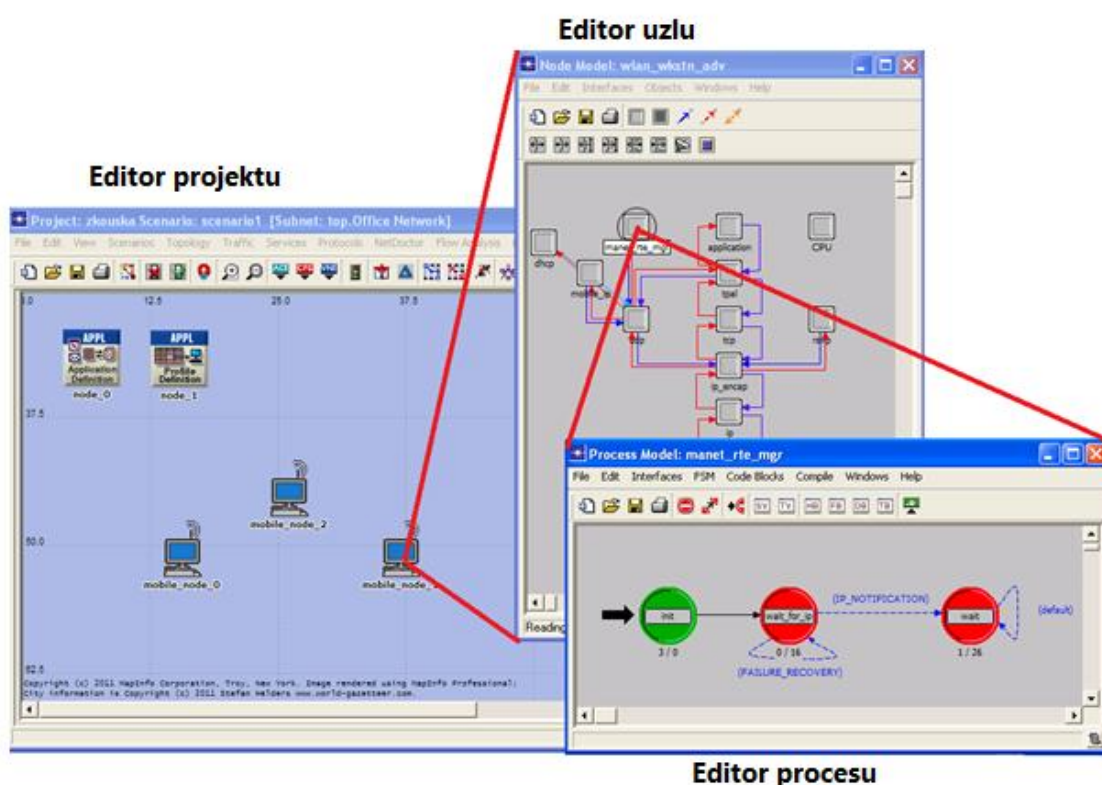
## 4 SIMULACE V PROGRAMU OPNET MODELER

### 4.1 OPNET Modeler

Program OPNET Modeler je simulační prostředí vyvinuté firmou OPNET Technologies Inc. Slouží pro návrh, simulaci a analýzu různých síťových technologií a mechanismů.[16]

#### 4.1.1 Editory

OPNET Modeler nabízí možnost použití velkého množství editorů, které umožňují vytvářet modely sítí a nastavovat jejich parametry s různým stupněm abstrakce. Ve většině případů se používají tři základní editory a to editor projektu, editor uzlu a editor procesů.



Obr. 4.1: Základní struktura editorů v OM.



#### **a) Editor projektu**

Je hlavní grafický editor modelující topologii a komunikaci v síti. Síť obsahuje jednotlivé uzly a odkazy na objekty. Projektový editor má v sobě implementovány mapy světa, díky nimž je struktura a fyzické rozložení sítě názornější. Samotný projekt sítě je pak tvořen jedním nebo více scénáři, které umožňují vytvoření odlišných konfigurací sítě v rámci jednoho projektu.[14]

#### **b) Editor uzlu**

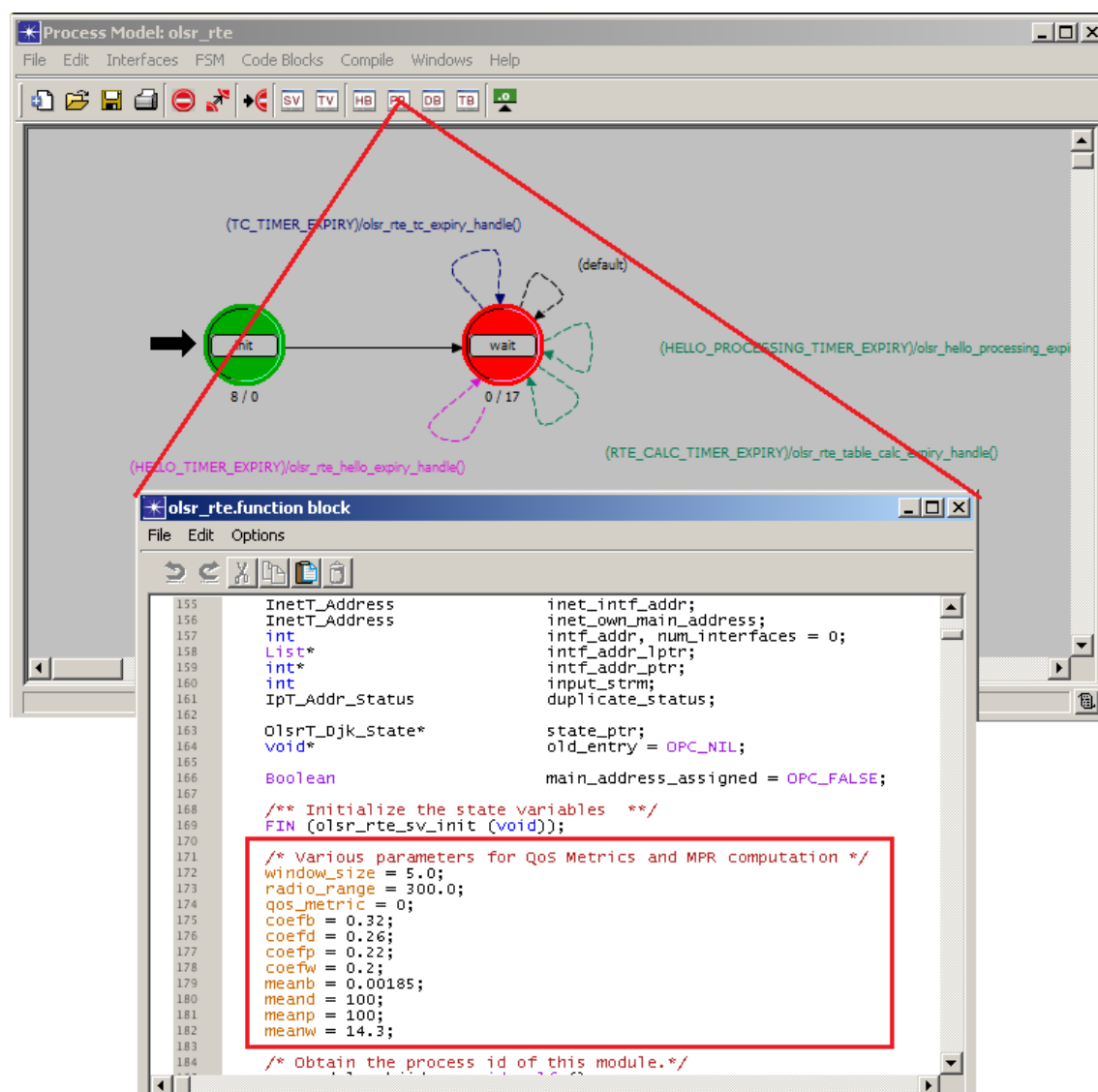
Představuje rozhraní nižší úrovně než projektový editor. Ukazuje architekturu síťového zařízení nebo systému a jeho vzájemné vztahy mezi funkčními modely a volanými funkcemi. Model uzlu se sestává z modulů uzlů, které jsou vzájemně propojeny datovými cestami a to buď tokem paketů, nebo statickými spoji. Každý modul může generovat, posílat a přijímat pakety od ostatních modulů uvnitř celého uzlu. Moduly typicky představují aplikace, protokolové vrstvy, algoritmy a fyzické prostředky jako jsou zásobníky, porty a sběrnice.[14]

#### **c) Editor procesu**

Procesní editor je konečný stavový automat přizpůsobený specifikovat všechny úrovně modelu. Stavby a přechody jsou definovány v grafickém diagramu. Každý stav a proces modelu obsahuje kód v jazyce C/C++ podporovaný rozsáhlou knihovnou s funkcemi vytvořenými pro protokolové programování. [14]

## 4.2 Model OLSR s podporou QoS

Ověření poskytování požadované kvality služeb v MANET sítích bylo provedeno pomocí simulace v programu OPNET Modeler. Pomocí poskytnutého modelu založeného na protokolu OLSR s podporou QoS byly provedeny navržené simulace.[4] Výsledky simulací jsou názorně zobrazeny v grafech a poté jsou jednotlivé dosažené výsledky podrobně popsány, viz kapitola 5.2. Pro nastavení požadovaných parametrů QoS bylo nutno nejprve nastavit parametry obsažené ve funkčním bloku modelu OLSR. Model a parametry jsou zobrazeny na obrázku č. 1.



Obrázek č. 4.2: Model protokolu OLSR a funkční blok modelu

Schéma procesního modelu OLSR s podporou QoS není oproti původnímu modelu OLSR obsaženém v OPNET Modeleru nijak pozměněno, a proto zde nebude rozebíráno. Ve funkčním bloku modelu oproti původnímu přibyla možnost nadefinovat určité QoS parametry. Na obrázku č.1 vidíme, že je možno nastavit parametry **window\_size** (velikost okna), **radio\_range** (rádiový dosah), **qos\_meric** (metrika QoS) a další pomocné parametry (**coefb**, **coefd**, **coefp**, **coefw**, **meanb**, **meand**), které slouží pouze pro pomocné výpočty a nebyly během simulace měněny, a proto zde nebudou blíže rozebírány. Parametr **window\_size** určuje, za jak dlouhou budou přepočítány QoS parametry. Parametr **radio\_range** určuje dosah stanice a je měřen v metrech. Poslední parametr **qos\_meric** může být nastaven na pět různých hodnot podle vlastních požadavků.

Hodnoty pro požadované QoS parametry jsou následující:

**Hodnota 0:** řídí se podle šířky pásma, ztrátovosti, zpoždění a ochotnosti

**Hodnota 1:** řídí se pouze podle šířky pásma

**Hodnota 2:** řídí se pouze podle zpoždění

**Hodnota 3:** řídí se pouze podle ztrátovosti

**Hodnota 4:** řídí se pouze podle ochotnosti

Ochotnost specifikuje to, jak ochotně bude uzel přebírat funkci přeposílání dat dalším uzlům na úkor ostatních okolních uzlů. Jak již bylo zmíněno dříve, tak parametry QoS jsou přepočítávány podle hodnoty nastavené v proměnné **window\_size**, pokud jsou používány standardní parametry Hello zpráv a TC zpráv, tak bude parametr **window\_size** nastaven na hodnotu 5. Dále bude parametr **radio\_range** nastaven na hodnotu 300, a parametr **qos\_meric** nastaven na hodnotu 0.

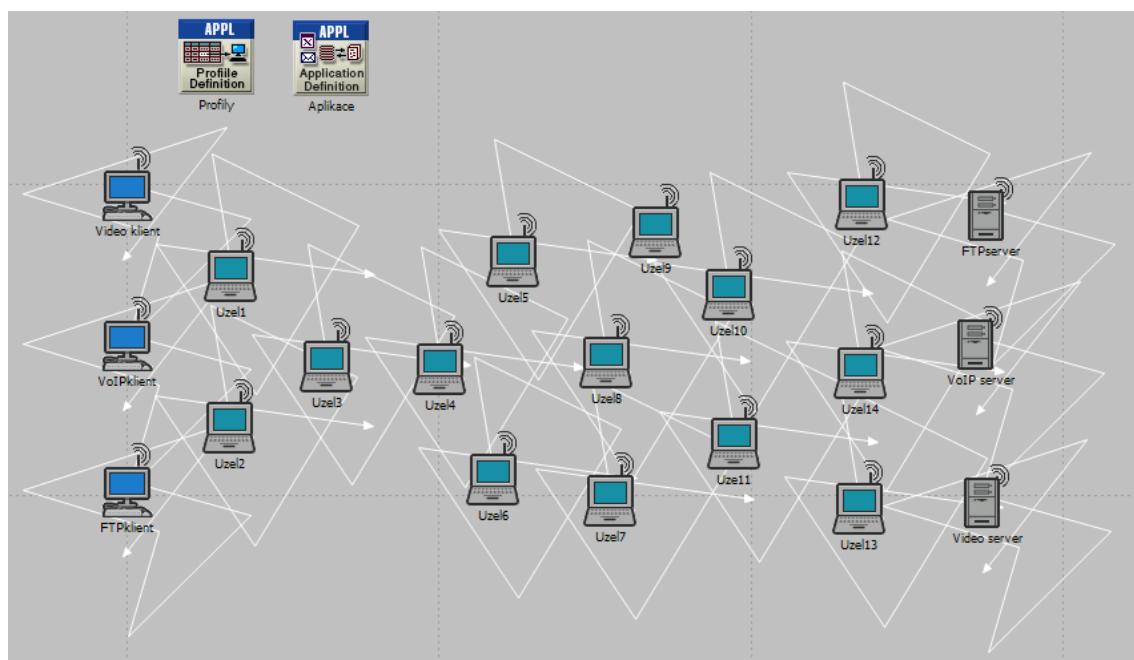
Nový procesní model OLSR obsahuje nová pole u HELLO a TC zpráv, tyto nová pole obsahují například parametry linek mezi uzly sítě, jako jsou zpoždění, ztrátovost nebo šířka pásma. Pomocí těchto parametrů je poté možno zaručit požadovanou kvalitu služeb v MANET sítích využívajících protokol OLSR

## 5 NÁVRH SIMULACE

V této kapitole bude rozebrána simulace MANET sítě využívající směrovací protokol OLSR. Úkolem simulace bylo ověřit podporu QoS u směrovacího protokolu OLSR u poskytnutého modelu. Tato kapitola bude rozdělena na 2 části. První část bude zaměřena na popis a návrh scénáře a modelové situace. Hlavním cílem druhé části bude rozbor výsledků simulace, jejich názorné grafické zobrazení a popis dosažených výsledků.

### 5.1 Návrh a popis modelové situace

Cílem tohoto scénáře je demonstrovat využití QoS v MANET sítích využívajících směrovací protokol OLSR. Scénář obsahuje 14 dynamických uzlů typu **manet\_station**, 3 uzly **wlan\_wkstn** a 3 uzly typu **wlan\_server**. Rozmístění jednotlivých uzlů je zobrazeno na obrázku 5.1.



Obrázek č. 5.1: Rozmístění uzlů

Tyto uzly se dynamicky pohybují po nastavených trajektoriích. Na stanicích **wlan\_wkstn** a **wlan\_server** označených jako **VoIP\_klient** a **VoIP\_server** je spuštěna hlasová služba, na stanicích označených jako **Video\_server** a **Video\_klient** je spuštěna služba videokonference a na stanicích označených jako **FTP\_server** a **FTP\_klient** je spuštěna FTP služba. U stanic **VoIP** (server, klient) a **Video** (server, klient) jsou spuštěny služby pracující s daty v reálném čase. U těchto dat je nutno zaručit určitou požadovanou kvalitu služeb, a proto právě tyto dvě služby budou hlavním cílem sledování u navržené simulace. Na zbylých 14 uzlech typu **manet\_station** je nastaven provoz na pozadí. Cílem sledování tedy bude, jak podpora QoS bude ovlivňovat kritické parametry potřebné pro doručení dat pracujících v reálném čase, mezi tyto parametry patří především zpoždění a kolísání zpoždění.

### 5.1.1 Návrh scénáře

#### Vytvoření projektu

Po spuštění programu OPNET Modeler byl vytvořen nový scénář. Pro vytvoření nového scénáře je třeba vybrat položku **File -> New** a zvolit **Project**. Po zadání nového názvu projektu a scénáře byla zvolena položka **Create empty scenario** a bylo zaškrtnuto použití **Startup Wizard**.

U specifikace oblasti byla zvolena možnost **Campus**, rozměry pracovní plochy byly nastaveny na hodnotu **3 x 3 Kilometers**. Dále byla zaškrtnuta v části **Model Family** položka **MANET** a hodnota **No** byla změněna na **YES**. Dokončení konfigurace bylo potvrzeno tlačítkem **Finish**.

#### Vytvoření modelu a jejich konfigurace

Z okna **Object Palette**, které se zobrazilo po dokončení konfigurace, bylo nejprve vybráno 14 stanic typu **manet\_station (Mobile Node)**, poté byly vybrány 3 stanice typu **wlan\_wkstn (Mobile Node)** a 3 stanice typu **wlan\_server (Mobile Node)**. Dále byly na pracovní plochu přetaženy položky **Application Config** a **Profile Config**, které jsou nutné pro nastavení parametrů jednotlivých aplikací, které budou v průběhu simulace spuštěny. Pro nastavení trajektorií jednotlivých uzlů bylo vybráno **Menu-> Topology -> Define Trajectory**.

Tato nově nadefinovaná trajektorie poté byla přiřazena k jednotlivým uzlům pomocí položky **Edit -> Attributes -> Trajectory** a byla vybrána požadovaná trajektorie. Dále byly přiřazeny IP (Internet Protocol) adresy jednotlivým uzlům v **Menu -> Protocols-> IP -> Addressing -> Auto-Assing IPv4 Addresses**. Pro jednotlivé uzly bylo nutno nastavit, který směrovací protokol budou využívat. V položce **Edit Attributes -> AD-HOC Routing Parameters -> AD-HOC Routing Protocol** byla zvolena u všech uzlů možnost **OLSR**.

### **Nastavení podpory QoS v MANET sítích v programu OPNET Modeler**

Podpora QoS byla zapnuta pomocí povolení standardu 802.11e a vybráním upraveného procesního modelu OLSR umožňujícího podporu QoS. Zapnutí standardu 802.11e se provede povolením HCF parametrů u každého bezdrátového uzlu sítě pomocí **Edit Atributtes -> Wireless Lan -> Wireless LAN Parameters -> HCF Parameters -> Status** změnou hodnoty na **Supported**. Procesní model OLSR s podporou QoS byl rozebrán v kapitole 4.2

### **Nastavení datového provozu a jeho záznamu**

U stanic typu **manet\_station** bylo vybráno **Edit Attributes -> MANET Trafficc Generation Parameters -> Number of Rows** (počet řádků) nastaveno na 1. Dále bylo potřeba nastavit hodnoty:

- **Start Time (s)** - na hodnotu intervalu **60**, to znamená, že vysílání dat začíná v definovaném čase.
- **Packet Size (bits)** - nastaveno na hodnotu constant **120000 bits** , každý paket bude mít tedy velikost 120 kbit.
- **Destination IP Address** – byla zvolena požadovaná IP adresa (např.: 192.0.1.8).
- **Packet Inter-Arrival Time (s)** – nastaven na hodnotu **20**, to znamená, že bude vyslán paket každou dvacátou vteřinu.

U stanic typu **wlan\_wkstn** a **wlan\_server** bylo nastavení provozu provedeno v objektech **Aplikace** a **Profily**. V objektu **Aplikace** bylo vybráno **Edit Attributes** -> **Application Definitions** -> **Number of Rows** a hodnota nastavena na **3**. První aplikace byla pojmenována **VoIPprovoz**, druhá **Videoprovoz** a třetí pak **FTPprovoz**.

Parametry jednotlivých aplikací jsou shrnuty v následujících tabulkách:

VoIP provoz			
Atribut	Hodnota	Atribut	Hodnota
Silence Length (seconds)	default	Compression Delay (seconds)	0.02
Talk Spurt Length (seconds)	default	Decompression Delay (seconds)	0.02
Symbolic Destination Name	Voice Destination	Conversation Environment	(...)
Encoder Scheme	G.711	Signaling	H.323
Voice Frames per Packet	1		
Type of Service	Interactive Voice (6)		
RSVP Parameters	None		
Traffic Mix (%)	All Discrete		

Tabulka č. 5.1: VoIP provoz

FTP provoz		Video provoz	
Atribut	Hodnota	Atribut	Hodnota
Command Mix (Get/Total)	100%	Frame Interarrival Time Information	10 frames/sec
Inter-Request Time (seconds)	constant (12)	Frame Size Information (bytes)	128X120 pixels
File Size (bytes)	constant (4000000)	Symbolic Destination Name	Video Destination
Symbolic Server Name	FTP Server	Type of Service	Streaming Multimedia (4)
Type of Service	Best Effort (0)	RSVP Parameters	None
RSVP Parameters	None	Traffic Mix (%)	All Discrete
Back-End Custom Application	Not Used		

Tabulka č. 5.2: Video a FTP provoz

U video a VoIP provozu byl nastaven požadovaný typ QoS. Video provoz při 10 snímcích za vteřinu a rozlišení 120 na 180 pixelů vytvoří datový tok o velikosti 340 kbit/s. U VoIP provozu byl nastaven kodek G.711, který má datový tok 16kbit/s. Provoz FTP byl nastaven tak, aby byl každých 10 vteřin zaslán soubor o velikosti 4 Mbit a typ QoS služby byl nastaven na best effort, to znamená, že se nejedná o aplikaci vyžadující QoS. U uzlů FTPklient, VoIPklient a VideoKlient byla spuštěna podpora požadovaných aplikací v položce Attributes -> Applications -> Supported Profiles. U uzlů FTPserver, VoIPserver a Videoserver byla spuštěna podpora požadovaných služeb v položce Attributes -> Applications -> Supported Services.

V objektu **Profily** poté bylo nastaveno, kdy se jednotlivé aplikace budou spouštět.

**Definované časy spouštění aplikací jsou následující:**

**a) Video provoz**

- Čas spuštění provozu (s): 180
- Čas ukončení provozu (s): 360

**b) VoIP provoz**

- Čas spuštění provozu (s): 10
- Čas ukončení provozu (s): 600

**c) FTP provoz**

- Čas spuštění provozu (s): 30
- Čas ukončení provozu (s): 430

**Výběr sledovaných charakteristik**

V Menu -> DES -> **Choose Individual Statistics** byly zvoleny požadované charakteristiky. V položce **Global Statistic** byly zvoleny možnosti **MANET, FTP, Voice** a **Video Conferencing**. V položce **Node Statistic** byly zvoleny možnosti **MANET, FTPserver, FTPklient Voice Application** a **Video Conferencing**.



## Spuštění a nastavení parametrů simulace

V Menu -> DES - > **Configure/Run Discrete Event Simulation** nebo stiskem ikony **Run** se objeví konfigurační okno simulace. Byly zvoleny následující parametry simulace:

- **Duration** - je doba sledování chování sítě, nastaveno na **10 minut**.
- **Values per statistic** - udává počet naměřených hodnot na vykreslení statistiky, byla zvolena hodnota **600**.
- **Simulation kernel** - definuje způsob překladač modelu do spustitelného kódu, bylo zvoleno **Optimized**.
- **Update Interval** - nastaven na hodnotu **10 000**.

Po dokončení simulace je nutno ukončit simulaci tlačítkem **Close**, nyní je možno výsledky simulace zobrazit.

## Zobrazení výsledků simulace

Pro zobrazení výsledků je nutno kliknout pravým tlačítkem myši a vybrat položku **View Results**. Pro grafické zobrazení jednotlivých statistik je nutno rozkliknout požadovanou položku, tedy **Global Statistic** nebo **Node Statistic** a zaškrtnout požadovaný pozorovaný parametr.

## 5.2 Zobrazení a vyhodnocení výsledků simulace

Program OPNET Modeler umožňuje sledovat nepřeberné množství parametrů sítě. Pro naše účely zde budou rozebrány pouze parametry důležité pro přenos dat aplikací, které pracují v reálném čase. Mezi tyto parametry patří především zpoždění a kolísání zpoždění neboli jitter.

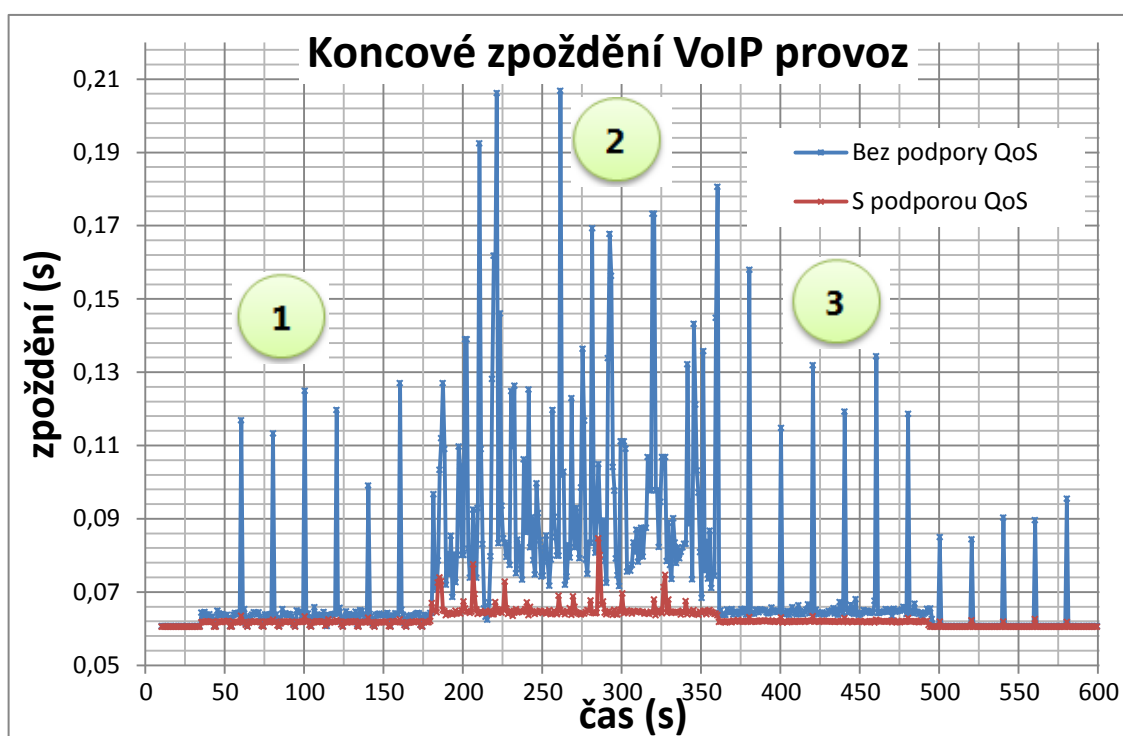
### 5.2.1 Koncové zpoždění

Zpoždění je definováno jako doba průchodu paketu sítí od jeho zdroje ke koncovému uzlu (příjemci dat). Krajiní mez hodnoty zpoždění, za kterou je hlasový hovor považován již nepoužitelný, je kolem 400ms. Nárůst zpoždění způsobují například prodlevy při soutěžení paketů nebo čekání ve frontě na vyslání v jednotlivých uzlech. Tento problém soutěžení a čekání ve frontě je řešen zajištěním kvality služeb. Pakety aplikací pracujících v reálném čase mají při implementaci kvality služeb větší prioritu než ostatní pakety. Na obrázku č. 1 je vidět, jak podpora QoS ovlivní zpoždění doručení paketů video provozu. Bez podpory QoS je průměrná hodnota zpoždění kolem 80ms, maximální hodnota je pak kolem 200ms. S podporou QoS je zpoždění mnohem nižší a pohybuje se kolem hodnoty 45ms.



Obrázek č. 5.2: Koncové zpoždění video provoz

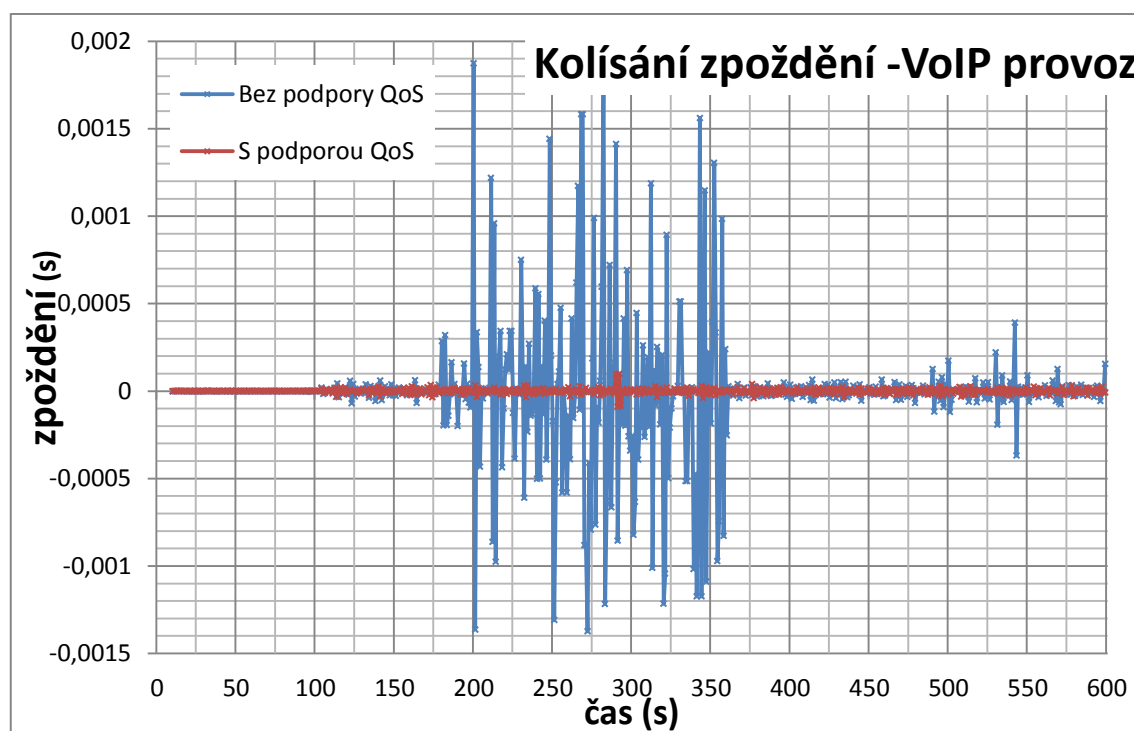
Na obrázku č. 2 je zobrazeno zpoždění pro hlasovou službu. V bodě **1** se drží hodnota zpoždění na nízké hodnotě, ale v případě, kdy není zaručena podpora QoS dochází k občasným navýšením hodnot zpoždění až na 120ms. V bodě **2** došlo ke spuštění video provozu a hodnota zpoždění se u scénáře bez podpory QoS vyhoupla i do hodnot přes 150ms, u scénáře s podporou QoS se hodnota zpoždění jen mírně zvýšila a drží se stále kolem hodnoty 70ms. Nakonec v bodě **3** došlo k ukončení video provozu a zpoždění se opět v případě scénáře s podporou QoS snížilo na hodnotu kolem 60ms, v případě kdy podpora QoS nebyla zaručena, se okamžité hodnoty zpoždění také ustálily, ale v některých časových okamžicích se opět dostávají až k hodnotám 120ms.



Obrázek č. 5.3: Koncové zpoždění video provozu

### 5.2.2 Kolísání zpoždění

Kolísání zpoždění je definováno jako rozdíl ve zpoždění doručovaných paketů. Hodnota kolísání zpoždění je velmi závislá na okamžitém vytížení sítě. Pokud hodnota kolísání zpoždění nepřesáhne hodnotu 50ms, pak je stále ještě možno provozovat na síti aplikace typu VoIP. Při realizaci podpory kvality služeb v síti je pak možno tento parametr snižovat tím, že jsou toky dat hlasových aplikací upřednostňovány před datovými toky ostatních aplikací.



Obrázek č. 5.4: Kolísání zpoždění VoIP provozu

Hodnota kolísání zpoždění není ani v jednom případě příliš vysoká. Ve scénáři, kde není zaručena kvalita služeb, se vyskytují větší výkyvy ve zpoždění, a to až k hodnotám kolem 1,5ms. V případě, kdy je kvalita služeb zaručena, jsou hodnoty kolísání zpoždění zanedbatelné. Oba scénáře jsou vhodné pro přenos VoIP služby.

### 5.2.3 Kvalita hovoru

Kvalita hovoru v síti je ohodnocena pomocí parametru **MOS** (Mean Opinion Score). Hodnoty MOS a popis, jak je hovor kvalitní je následující: [17]

**MOS 5** – vynikající kvalita hovoru, srovnatelná s hovorem mezi lidmi v malé vzdálenosti

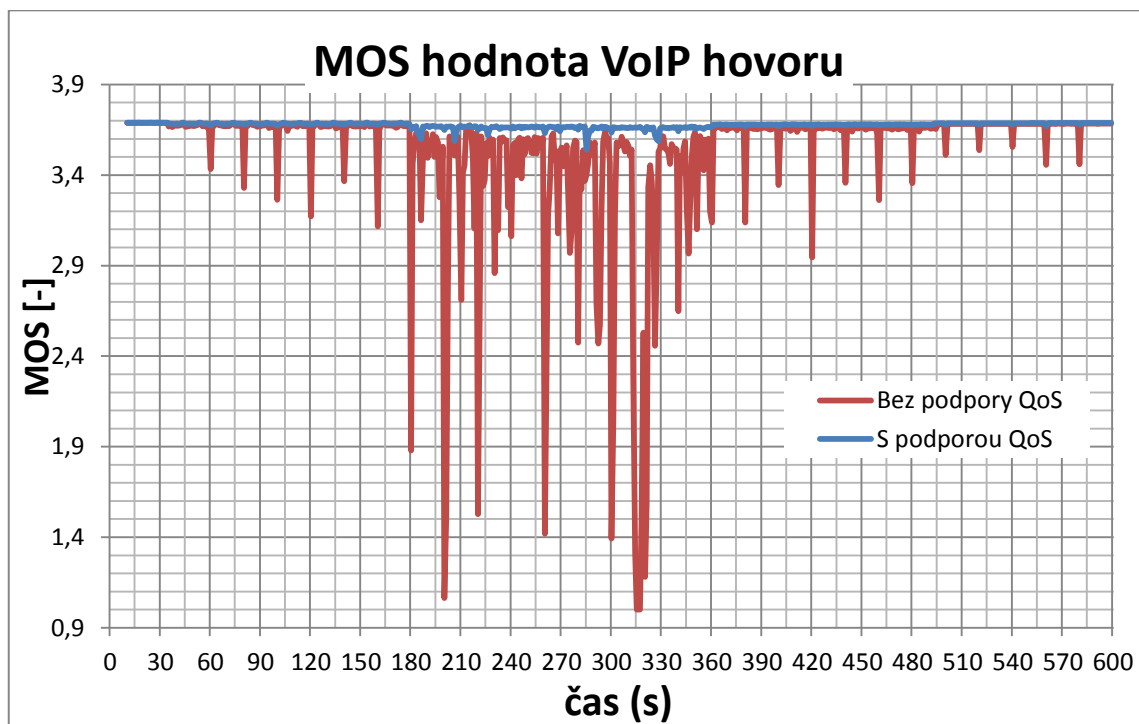
**MOS 4** – u hovoru se objevuje mírné rušení

**MOS 3** – průměrná kvalita hovoru, rušení je znatelnější

**MOS 2** – hovoru jde už špatně rozumět, velké rušení

**MOS 1** – není možno vést komunikaci

Na obrázku č. 4 je znázorněna MOS hodnota VoIP hovoru. V případě, kdy nebyla zapnuta podpora QoS, jsou zřetelné výrazné výpadky kvality hovoru při spuštění video provozu, hodnota MOS klesá až k hodnotám pod 2, toto má za následek nepříjemné rušení v průběhu hovoru, k těmto výpadkům kvality hovoru dochází z důvodu velkého zpoždění paketů nebo příliš velkého kolísání zpoždění. V případě podpory QoS je změna hodnoty MOS zanedbatelná i při spuštění video provozu.



Obrázek č. 5.5: Hodnota MOS

V tabulce č. 1,2 a 3. jsou porovnány maximální, minimální a průměrné hodnoty zpoždění a MOS hodnoty pro scénář bez podpory QoS a pro scénář s podporou QoS.

<b>MOS</b>	<b>Bez podpory QoS</b>	<b>S podporou QoS</b>
Největší hodnota (-)	3,688	3,689
Nejmenší hodnota (-)	1,000	3,539
Průměrná hodnota (-)	3,531	3,675

Tabulka č. 5.3: MOS

<b>Zpoždění video vysílání</b>	<b>Bez podpory QoS</b>	<b>S podporou QoS</b>
Největší hodnota (s)	0,06199	0,080766
Nejmenší hodnota (s)	0,22439	0,032094
Průměrná hodnota (s)	0,091891	0,039481

Tabulka č. 5.4: Zpoždění video vysílání

<b>Zpoždění VoIP vysílání</b>	<b>Bez podpory QoS</b>	<b>S podporou QoS</b>
Největší hodnota (s)	0,206781	0,08463
Nejmenší hodnota (s)	0,060618	0,060517
Průměrná hodnota (s)	0,074258	0,062601

Tabulka č. 5.5: Zpoždění VoIP vysílání

## 6 IMPLEMENTACE SMĚROVACÍCH PROTOKOLŮ

V předchozích kapitolách byly rozebrány teoretické poznatky o směrovacích protokolech pro MANET sítě a jak je u nich možno zajistit podporu požadované kvality služeb. Tato kapitola se bude zabývat zavedením směrovacích protokolů do reálné MANET sítě. Budou zde rozebrány poznatky o existujících implementacích směrovacích protokolů a stav podpory požadované kvality služeb u uvedených směrovacích protokolů.

### 6.1 Stav implementací směrovacích protokolů

Existuje velké množství implementací nejen pro nasazení do reálné MANET sítě ale i do různých simulačních nástrojů jako jsou Network Simulator 2 (NS 2) nebo OPNET Modeler. Většina implementací pro reálné MANET sítě je navržena pro operační systém Linux a je dostupná jako open-source projekt. Budou zde vyjmenovány a krátce popsány některé možné implementace právě pro operační systém Linux. Některé implementace jsou upravovány už několik let a jsou ověřeny pro nasazení do reálné MANET sítě, některé z implementací jsou teprve v počátcích vývoje a nejsou vhodné pro nasazení do reálných MANET sítí.

#### **Implementace AODV-UU:**

Tato implementace protokolu AODV je vyvíjena na Univerzitě Uppsala, podporuje IPv6 a multicast. Není zde použita žádná podpora kvality služeb. [2]

#### **Implementace UniK OLSR Daemon**

Jedná se o ověřenou a kvalitní implementaci protokolu OLSR, kompatibilní s RFC3626. Neobsahuje žádnou podporu kvality služeb. Tato implementace je už aplikována v reálných MANET sítích některých výzkumných organizací, kde je počet uzlů i kolem 1000. Výpočetní náročnost na procesor je u této implementace velmi malá, což je výhoda pro uzly, které nemají tak kvalitní procesorovou jednotku. [23]

## **Implementace BABEL**

Tato implementace směrovacího protokolu BABEL podporuje IPv6. Tento protokol je navržen, tak aby byl spolehlivý jak na drátových sítích, tak i na bezdrátových mesh sítích. Implementace nepodporuje žádnou kvalitu služeb. [3]

## **Implementace QOLYESTER**

Jedné se o implementaci protokolu Q-OLSR, tedy protokolu OLSR s podporou QoS. Implementace je kompatibilní podle dokumentu RFC3626. Podpora QoS je bohužel pouze ve stadiu vývoje a není zde ještě implementována. Podle [18] bude podpora kvality služeb přidána, jakmile bude vytvořena použitelná implementace a řádně otestována.

## **Implementace DSR**

Tato implementace protokolu DSR je ve stádiu vývoje a není považována za stabilní. Neobsahuje žádnou podporu kvality služeb. [11]

Z následujícího výčtu možných implementací směrovacích protokolů pro nasazení do reálné MANET sítě, je možno vyvodit, že je podpora směrování v reálných MANET sítích stále ve vývoji a není zde implementována žádná podpora kvality služeb nebo je teprve v návrhu pro další vývoj. Je tedy velkou výzvou zajistit u některých výše zmíněných implementací požadovanou kvalitu služeb. Je třeba zmínit, že existují implementace podpory kvality služeb u směrovacích protokolů AODV a OLSR ale pouze pro simulační nástroj OPNET Modeler, to znamená, že zatím není možno tuto podporu nasadit přímo do reálné MANET sítě. Ověření funkčnosti podpory kvality služeb u směrovacího protokolu OLSR pro simulační nástroj OPNET Modeler bylo rozebráno v kapitole 5.



## 6.2 Nasazení protokolu OLSR do reálné MANET sítě

Jak už bylo zmíněno v kapitole 2.4, tak protokol OLSR se vyznačuje tím, že se jedná o proaktivní směrovací protokol, a proto si každý uzel uchovává veškeré informace o topologii sítě. Důsledkem tohoto je, že protokol OLSR má velmi malé směrovací záhlaví a velmi nízké zpoždění v síti. Ze studie [7], kde byly prováděny simulace v nástroji NS2 se směrovacími protokoly AODV, DSDV, DSR a OLSR lze vypožorovat, že protokol OLSR má podle předpokladů nižší zpoždění a nižší kolísání zpoždění než protokoly AODV a DSR. Nevýhoda protokolu OLSR je naopak jeho nižší propustnost oproti protokolům AODV nebo DSR. Pro realizaci reálné MANET sítě byl nakonec zvolen protokol OLSR. Důvodem zvolení protokolu OLSR jsou především jeho nízké hodnoty zpoždění i při větším zatížení sítě a při větší pohyblivosti uzlů, což je vhodné pro MANET sítě, kde jsou přenášena hlasová data, u kterých je důležité, aby se hodnoty zpoždění a kolísání zpoždění držely na nízkých hodnotách. Protokol OLSR byl také zvolen proto, že se jedná o ověřený směrovací protokol, který je už nyní nasazen pro práci v reálných podmínkách, kde ho využívají některé výzkumné organizace.

Pro další práci, návrh a realizaci reálné MANET sítě se směrovacím protokolem OLSR byla zvolena implementace **UniK OLSR** pro operační systém Linux. Jedná se o stabilní a stále vyvíjenou implementaci protokolu OLSR. Tato implementace podporuje přidávání zásuvných modulů, a je proto do této implementace možné relativně jednoduše přidávat další funkce, jako je například podpora kvality služeb.

## 7 REALIZACE REÁLNÉ MANET SÍTĚ

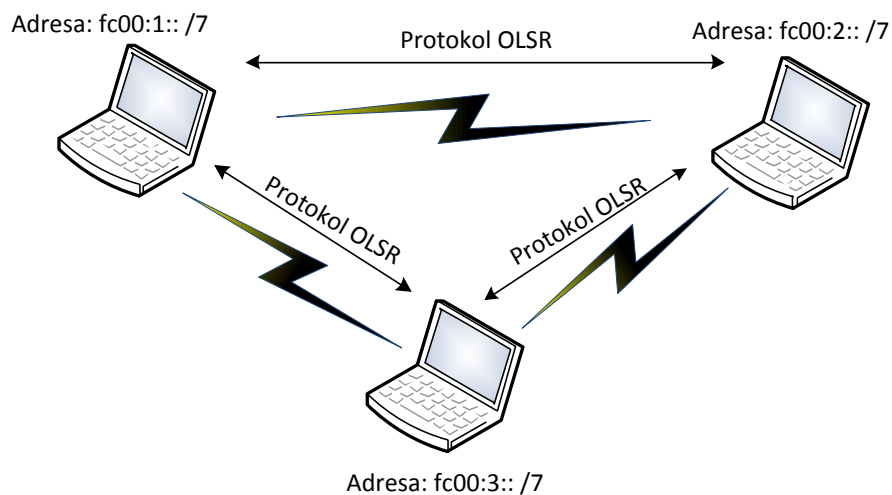
### 7.1 Základní návrh realizace

Pro realizaci reálné MANET sítě byly zvoleny dva počítače s bezdrátovou síťovou kartou, na kterých byla spuštěna linuxová distribuce Ubuntu 11.10 a na kterých byl nainstalován a spuštěn OLSR démon, pomocí něhož síťová komunikace na transportní vrstvě probíhá právě s využitím směrovacího protokolu OLSR.

#### Použité vybavení a programy:

- 2x Notebook Lenovo ThinkPad X201
- Notebook Acer Aspire 5110
- Linuxová distribuce Ubuntu 11.10 [6]
- Doplněk pro podporu protokolu OLSR : OLSR démon [23]
- Program pro zachycení síťové komunikace Wireshark

#### Základní schéma realizace:



Obr. č. 7.1: Schéma realizace reálné MANET sítě

## 7.2 OLSR démon

### 7.2.1 Instalace OLSR démona

Pro zprovoznění komunikace založené na protokolu OLSR je potřeba několika kroků. Nejprve je potřeba stáhnout a nainstalovat programy **bison**, **flex** a **git**. Pro instalaci těchto programů je potřeba spustit terminál a napsat níže uvedené příkazy. Instalace a konfigurace OLSR démona je potřebná na všech počítačích, které mezi sebou komunikují pomocí protokolu OLSR.

```
# sudo apt-get install bison
```

```
# sudo apt-get install flex
```

```
# sudo apt-get install git
```

Nyní je potřeba si vytvořit v domovském adresáři (/home) složku OLSR pomocí příkazu:

```
# sudo mkdir OLSR
```

Dále je potřeba přepnout se do nově vytvořené složky OLSR příkazem:

```
# cd OLSR
```

Následujícím příkazem se provede stažení aktuální verze OLSR démona:

```
# git clone http://olsr.org/git/olsrd.git
```

Poté se je nutno přepnout do nově vytvořené složky OLSRd ve složce OLSR pomocí příkazu:

```
# cd OLSRd
```

Na závěr ještě napíšeme následující dva příkazy a instalace je hotova:

```
# sudo make
```

```
# sudo make install
```

### 7.2.2 Konfigurace OLSR démona

Konfigurace se provádí v konfiguračním souboru *olsrd.conf*. Pro změnu konfiguračního souboru pomocí programu **nano**, je potřeba zadat následující příkaz:

```
# sudo nano /etc/olsrd.conf
```

Konfigurační soubor se skládá z několika částí. Níže bude rozebráno, jaké jsou základní části konfiguračního souboru, a bude vždy tučně uvedeno, které příkazy byly v konfiguračním souboru dopsány nebo upraveny.

#### a) Základní konfigurace:

Zde jsou uloženy pouze základní informace, je možno nastavit úroveň ladění (0-9 ) a typ internetového protokolu (IPv4 nebo IPv6).

***DebugLevel 1***

***IpVersion 6***

#### b) Konfigurace OLSR agenta:

Tyto parametry kontrolují nastavení směrovacího agenta, který není spjat s OLSR protokolem nebo jeho rozšířením.

#### c) Specifické linuxové rozšíření OLSR démona

#### d) Nastavení démona OLSR protokolu:

Zde je možno nastavit síťovou asociaci (HNA – Host network association). Toto umožňuje protokolu OLSR oznamovat přídavné IP adresy nebo IP podsítě, které jsou přes tento uzel dostupné. Pokud je potřeba použít bránu, například pro přístup do Internetu, pak je potřeba využít právě této síťové asociace. Příklad nastavení brány v konfiguračním souboru *olsrd.conf* může být následující:

***HNA6 :: 0***

**e) Nastavení rozšíření protokolu OLSR**

**f) Konfigurace zásuvných modulů:**

Zde je možno nastavit zásuvné moduly pro OLSR démona. Mezi zásuvné moduly patří například použitý modul **txtinfo**.

```
LoadPlugin "olsrd_txtinfo.so.0.1
```

```
{
```

```
  PIParam "port" "8080"
```

```
  PIParam "Host" "127.0.0.1"
```

```
}
```

**g) Výchozí konfigurace rozhraní OLSR démona**

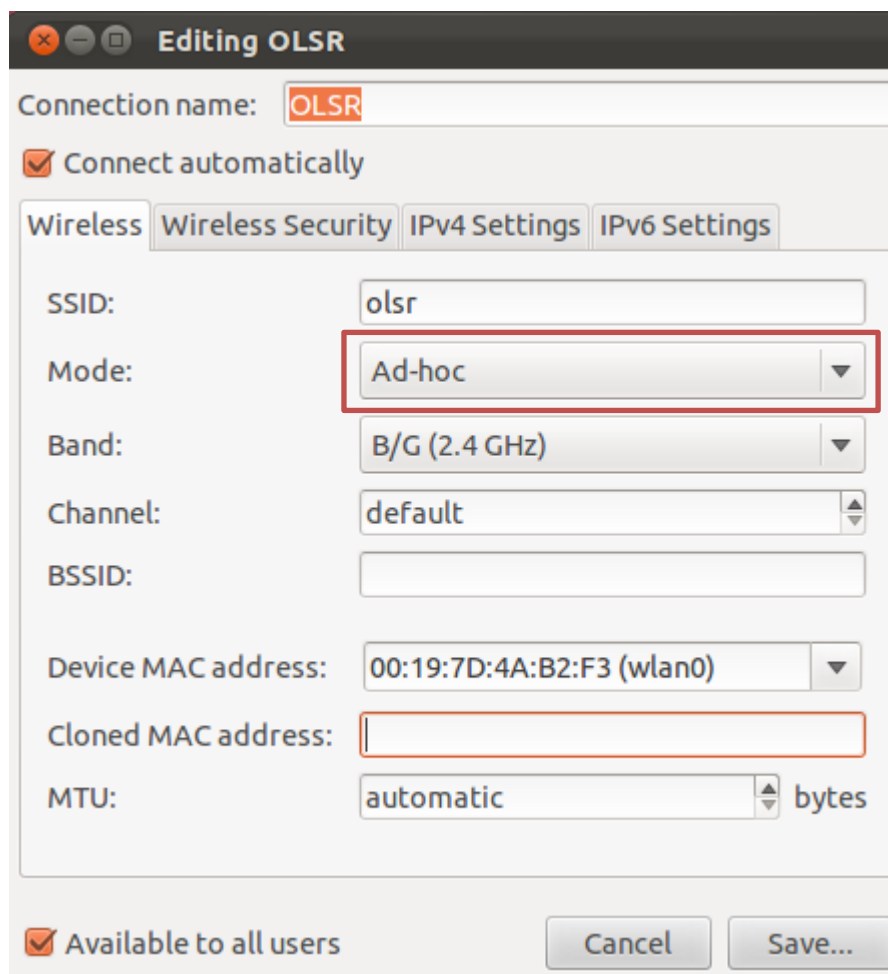
Pokud existuje více síťových rozhraní, pak je možno v tomto bloku definovat určité parametry pro všechny síťová rozhraní najednou.

**h) Konfigurace OLSRd rozhraní**

Do tohoto konfiguračního bloku se přidávají jednotlivá síťová rozhraní. Je možno uvést i více než jedno podporované síťové rozhraní.

```
Interface "wlan0"
```

Před spuštěním OLSR démona je potřeba vytvořit bezdrátovou Ad-hoc síť. Vytvoření a nastavení je zobrazeno na obrázku č. 7.2.



Obrázek č. 7.2: Vytvoření a nastavení bezdrátové Ad-hoc sítě

Je důležité změnit mód fungování bezdrátové karty na typ **Ad-hoc**.

### 7.2.3 Spuštění OLSR démona

Spuštění OLSR démona a tedy i síťové komunikace s podporou protokolu OLSR se provede pomocí následujícího příkazu: `# sudo olsrd`

Na následujícím obrázku vidíme výpis základní konfigurace OLSR démona po jeho spuštění.

```
*** olsr.org - pre-0.6.3-git_0081402-hash_36723d6e66f8ac59d49044c072135275 ***
Build date: 2012-02-10 14:18:16 on roman-Aspire-5110
http://www.olsr.org

Parsing file: "/etc/olsrd.conf"
Debug level: 1
IpVersion: 6
Use NIIT ip translation: enabled
LQ Algorithm: etx_ff
Link quality fish eye 0

Interface Defaultssetting ifs_in_curr_cfg = 0
  IPv4 broadcast/multicast : AUTO (d)
  Mode : mesh (d)
  IPv6 multicast : ff02::6d
  HELLO emission/validity : 2.00 (d)/20.00 (d)
  TC emission/validity : 5.00 (d)/300.00 (d)
  MID emission/validity : 5.00 (d)/300.00 (d)
  HNA emission/validity : 5.00 (d)/300.00 (d)
  Autodetect changes : yes
  IPv4 broadcast/multicast : AUTO
  Mode : mesh
  IPv6 multicast : ::
  HELLO emission/validity : 0.00/0.00
  TC emission/validity : 0.00/0.00
  MID emission/validity : 0.00/0.00
  HNA emission/validity : 0.00/0.00
  Autodetect changes : no
Warning, niit4to6 device is not available, deactivating NIIT
Added :: to IP deny set
Added ::1 to IP deny set

---- Interface configuration ----
```

Obrázek č.7.3: Základní konfigurace

- 1 Výpis verze použité implementace a čas kompilace dané implementace
- 2 Výpis základních parametrů nastavených v konfiguračním souboru `olsrd.conf` viz kapitola 7.2.2.
- 3 Výpis výchozího nastavení rozhraní a použití a frekvence vysíláních paketů

Na obrázku č.7.4 je zobrazena konfigurace bezdrátového rozhraní

```
roman@roman-Aspire-5110: ~  
---- Interface configuration ----  
Checking wlan0:  
Wireless interface detected  
Address: ff02::6d  
Multicast: ff02::6d  
Metric: 1  
MTU - IPHdr: 1438  
Index 3  
Address: fc00:1::  
Multicast: ff02::6d  
Join multicast send: Address already in use  
Join multicast send: Address already in use  
New main address: fc00:1::  
Using 'etx_ff' algorithm for lq calculation.  
TC: add entry fc00:1::  
RIB: add prefix fc00:1::/128 from fc00:1::  
-- ALL PLUGINS LOADED! --  
Main address: fc00:1::  
Scheduler started - polling every 0.050000 ms  
  
*** olsr.org - pre-0.6.3-git_0081402-hash_36723d6e66f8ac59d49044c0721  
75 - (2012-02-10 14:18:16 on roman-Aspire-5110) ***  
  
--- 09:23:38.941828 ----- LINK  
  
IP address          hyst      LQ      ETX  
--- 09:23:38.941973 ----- TWO-HOP NEIGHBORS  
  
IP addr (2-hop)  IP addr (1-hop)  Total cost  
||
```

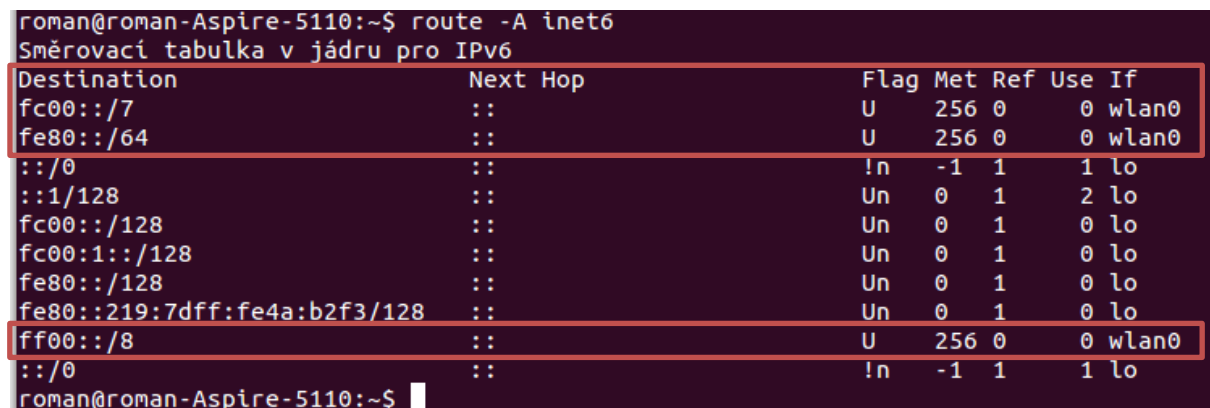
Obrázek č. 7.4: Konfigurace rozhraní

- 1 Jedná se o výpis základního nastavení rozhraní **wlan0** , rozhraní je nastaveno tak, aby využívalo protokol IPv6. Je zde také nastavena podpora multicastu.
- 2 Nastavení adresy, která bude použita jako hlavní (výchozí) adresa, je převzata z manuálního nastavení pevné IPv6 adresy u bezdrátového rozhraní
- 3 Výpis adres, přes které jsou ostatní zařízení připojena k danému uzlu. Výpis je zatím prázdný, způsob přidávání ostatních uzlů je nastíněn v následující kapitole



Na obrázku č. 7.5 můžeme pomocí následujícího příkazu vypsat směrovací tabulku, ve které je možno nalézt nový záznam

```
# sudo route -A inet6
```



```
roman@roman-Aspire-5110:~$ route -A inet6
Směrovací tabulka v jádru pro IPv6
```

Destination	Next Hop	Flag	Met	Ref	Use	If
fc00::/7	::	U	256	0	0	wlan0
fe80::/64	::	U	256	0	0	wlan0
::/0	::	!n	-1	1	1	lo
::1/128	::	Un	0	1	2	lo
fc00::/128	::	Un	0	1	0	lo
fc00:1::/128	::	Un	0	1	0	lo
fe80::/128	::	Un	0	1	0	lo
fe80::219:7dff:fe4a:b2f3/128	::	Un	0	1	0	lo
ff00::/8	::	U	256	0	0	wlan0
::/0	::	!n	-1	1	1	lo

```
roman@roman-Aspire-5110:~$
```

Obrázek č. 7.5: Výpis směrovací tabulky

Do směrovací tabulky byly po spuštění OLSR démona přidány nové adresy pro rozhraní wlan0.

Nebudou zde dále zobrazeny nastavení ostatních počítačů a jejich směrovacích tabulek, jelikož nastavení je analogické, jen je přiřazena jiná IPv6 adresa podle toho o jaký počítač se jedná.

#### 7.2.4 Přidávání nových uzlů

Použití pouze pro jeden uzel je samozřejmě zbytečné, proto je možno další uzly možno detekovat a přidat do směrovacích tabulek. Přidání ostatních uzlu je automatické, přidání ostatních hostů se děje na základě výměn OLSR zpráv mezi počítači.

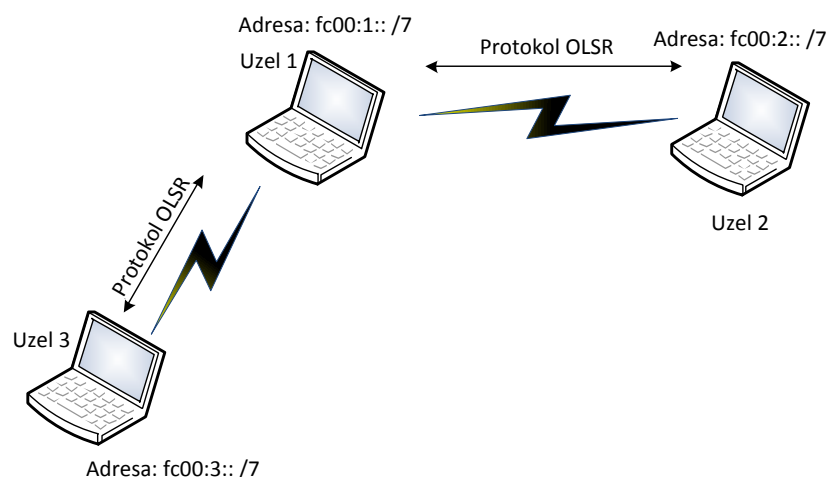
```
roman@roman-Aspire-5110: ~  
*** olsr.org - pre-0.6.3-git_0081402-hash_36723d6e66f8ac59d49044c0721352  
75 - (2012-02-28 18:52:31 on roman-Aspire-5110) ***  
  
--- 12:39:12.271632 ----- LINKS  
  
IP address          hyst      LQ      ETX  
fc00:3::            0.000    1.000/1.000  1.000  
fc00:2::            0.000    0.944/1.000  1.058  
  
--- 12:39:12.271732 ----- TWO-HOP NEIGHBORS  
  
IP addr (2-hop)    IP addr (1-hop)    Total cost  
fc00:3::          fc00:2::          2.176  
fc00:2::          fc00:3::          2.118
```

Obrázek č. 7.6: Tabulka hostů připojených na daný počítač

- 1** Seznam IPv6 adres hostů připojených k danému počítači.
- 2** Seznam IPv6 adres hostů, kteří jsou dostupní od daného uzlu na vzdálenost 2 skoků.
- 3** Seznam IPv6 adres hostů, kteří jsou dostupní od daného uzlu na vzdálenost 1 skoku, to znamená, že jsou propojeni přímo, bez mezilehlého uzlu.

Opět je zde zobrazeno přidávání hostů jen na jednom počítači, jelikož situace je pro ostatní počítače analogická, rozdíl bude jen v adresách připojených hostů.

Následující případ zobrazený na obrázku č. 7.7 vystihuje podstatu mezilehlého uzlu. Uzel 3 je vzdálen od uzlu 2 tak daleko, že není mezi nimi možná přímá komunikace, a proto je nutno zajistit komunikace mezi těmito uzly právě pomocí mezilehlého uzlu, který daný provoz směruje dále v síti.



Obrázek č. 7.7: Scénář s komunikací pomocí mezilehlého uzlu

Tento scénář, kdy je uzel 3 vzdálen od uzlu 2 tak daleko, že není mezi nimi možná přímá komunikace, je možno zajistit zjednodušeně následujícím zásahem do směrovací tabulky, není přitom nutno počítače mezi sebou fyzicky vzdalovat. Využijeme k tomu příkazu **ip6tables**:

```
# ip6tables -A INPUT -m mac --mac-source XX:XX:XX:XX:XX:XX -j DROP
```

Za argument `XX:XX:XX:XX:XX:XX` je potřeba dosadit fyzickou adresu bezdrátové karty počítače. Pokud podle scénáře na obrázku č. 7.8 nechceme z uzlu 3 přímo komunikovat s uzlem 2, pak musíme na uzlu 3 napsat výše uvedený příkaz do příkazové řádky a jako fyzickou adresu uvedeme fyzickou adresu bezdrátového rozhraní uzlu 2. Tento příkaz vlastně docílí toho, že budou zahazovány všechny pakety, které budou přicházet od uzlu, se kterým má být docílena komunikace pouze přes mezilehlý uzel.

Destination	Next Hop	Flag	Met	Ref	Use	If
fc00:3::/128	fc00:1::	U	2	0	0	wlan0

Obrázek č. 7.8: Směrovací tabulka s mezilehlým uzlem

Next Hop neboli další skok označuje, kam budou odesílány pakety, pokud je cílová adresa (Destination) uzel 3.

### 7.2.5 Podpora zásuvných modulů

Pomocí těchto modulů je možno doplňovat některé nové funkce k základní implementaci protokolu OLSR. Jeden z těchto modulů je například modul, který směřuje pakety v síti pomocí stavu baterie jednotlivých uzlů. Doplnků je velké množství a lze je nalézt ve složce **lib** na uvedeném místě: `/home/olsr/olsrd/lib`

### 7.2.6 Brána do jiných sítí

MANET síť založenou na OLSR protokolu je možno propojit s ostatními sítěmi, například ethernetovou sítí, pomocí něhož může být MANET síť napojena do sítě Internet. K tomuto účelu slouží zprávy typu HNA – Host and Network Association. Tyto zprávy jsou zasílány v pravidelných intervalech pomocí bran (gateway). Pokud je potřeba, aby daný uzel sloužil jako brána, je nutno provést nastavení v konfiguračním souboru **olsrd.conf**. Příklad nastavení brány v konfiguračním souboru **olsrd.conf** je popsán v kapitole 7.2.2.

Význam zápisu IPv6 adresy „`::/0`“ označuje, že se jedná o připojení do sítě Internet.

Směrovací tabulka hosta, jenž neslouží jako brána, je zobrazena na obrázku č. 7.9.

Destination	Next Hop	Flag	Met	Ref	Use	If
fc00:1::/128	::	U	2	0	1	wlan0
fc00:3::/128	::	U	2	0	4	wlan0
fc00::/7	::	U	256	0	0	wlan0
fe80::/64	::	U	256	0	0	wlan0
::/0	fc00:1::	UG	2	0	0	wlan0
::/0	::	!n	-1	1	38	lo
::1/128	::	Un	0	1	1	lo
fc00::/128	::	Un	0	1	0	lo
fc00:2::/128	::	Un	0	1	0	lo
fe80::/128	::	Un	0	1	0	lo
fe80::227:10ff:feda:4358/128	::	Un	0	1	0	lo
ff00::/8	::	U	256	0	0	wlan0
::/0	::	!n	-1	1	38	lo

Obrázek č. 7.9: Směrovací tabulka – pouze adresy IPv6

Příznak UG v položce FLAG značí, že se jedná o funkci brány.

## 8 ANALÝZA SÍŤOVÉ KOMUNIKACE

Pro zachycení a analýzu síťové komunikace MANET sítě využívající protokol OLSR byl zvolen program Wireshark. Program Wireshark lze spustit pomocí následujícího příkazu:

```
# sudo wireshark, pokud není nainstalován tak je ho nutno nainstalovat pomocí příkazu:
```

```
# sudo apt-get install wireshark
```

### 8.1 Analýza OLSR paketů

V rámci sítě využívající protokol OLSR dochází k pravidelným výměnám HELLO a TC zpráv, viz kapitola 2.4. Zachycení OLSR paketů obsahujících HELLO nebo TC zprávu je zobrazen na obrázku č. 8.1. Paket OLSR může obsahovat buď pouze zprávu HELLO, nebo může být připojena také TC zpráva, podle toho zda je připojena i TC zpráva se odvíjí i velikost OLSR paketu. OLSR paket bez TC zprávy má délku 144 bajtů a paket bez zprávy TC má pouze 76 bajtů. Všechny OLSR pakety jsou zasílány na multicastovou adresu ff02::6d [19]. Ještě je možno zmínit, že protokol OLSR využívá pro přenos dat transportní protokol UDP - port 698.

No.	Time	Source	Destination	Protocol	Length	Info
69	39.500629	fc00:3::	ff02::6d	OLSR v1	138	OLSR (IPv6) Packet, Length: 76 Bytes
70	39.842211	fc00:2::	ff02::6d	OLSR v1	206	OLSR (IPv6) Packet, Length: 144 Bytes
71	40.615972	fc00:1::	ff02::6d	OLSR v1	206	OLSR (IPv6) Packet, Length: 144 Bytes
72	41.408838	fc00:3::	ff02::6d	OLSR v1	206	OLSR (IPv6) Packet, Length: 144 Bytes
73	41.599195	fc00:2::	ff02::6d	OLSR v1	138	OLSR (IPv6) Packet, Length: 76 Bytes
74	42.471428	fc00:1::	ff02::6d	OLSR v1	138	OLSR (IPv6) Packet, Length: 76 Bytes
75	43.153031	fc00:2::	ff02::6d	OLSR v1	206	OLSR (IPv6) Packet, Length: 144 Bytes

▶ Frame 3: 206 bytes on wire (1648 bits), 206 bytes captured (1648 bits)

▶ Ethernet II, Src: IntelCor 55:86:18 (00:27:10:55:86:18), Dst: IPv6mcast 00:00:00:6d (33:33:00:00:00:6d)

▶ Internet Protocol Version 6, Src: fc00:3:: (fc00:3::), Dst: ff02::6d (ff02::6d)

▼ User Datagram Protocol, Src Port: olsr (698), Dst Port: olsr (698)

Source port: olsr (698)  
Destination port: olsr (698)  
Length: 152  
Checksum: 0x8c0b [validation disabled]

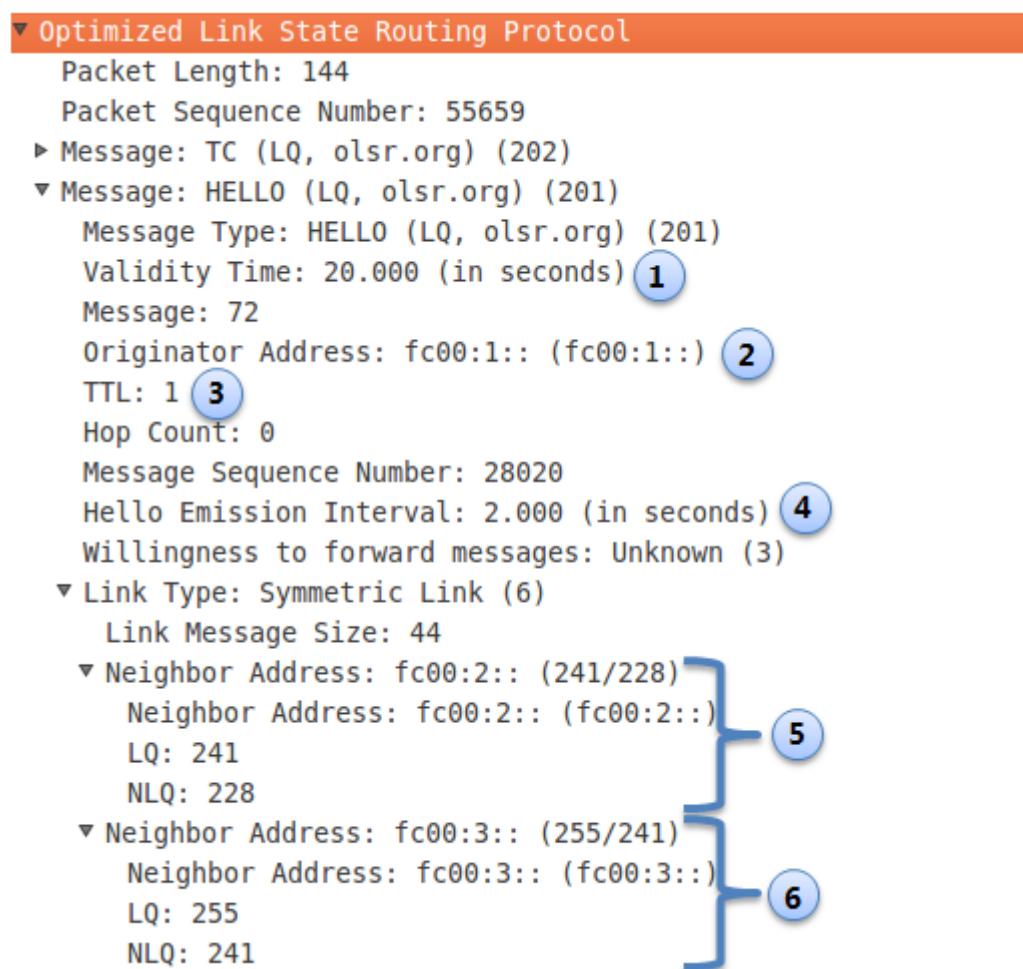
▼ Optimized Link State Routing Protocol

Packet Length: 144  
Packet Sequence Number: 61888  
▶ Message: TC (LQ, olsr.org) (202)  
▶ Message: HELLO (LQ, olsr.org) (201)

Obrázek č. 8.1 : Zachycení OLSR paketů programem Wireshark

### 8.1.1 Analýza HELLO zprávy

Obrázek č. 8.2 zobrazuje detailní výpis HELLO zprávy zachycené programem Wireshark.

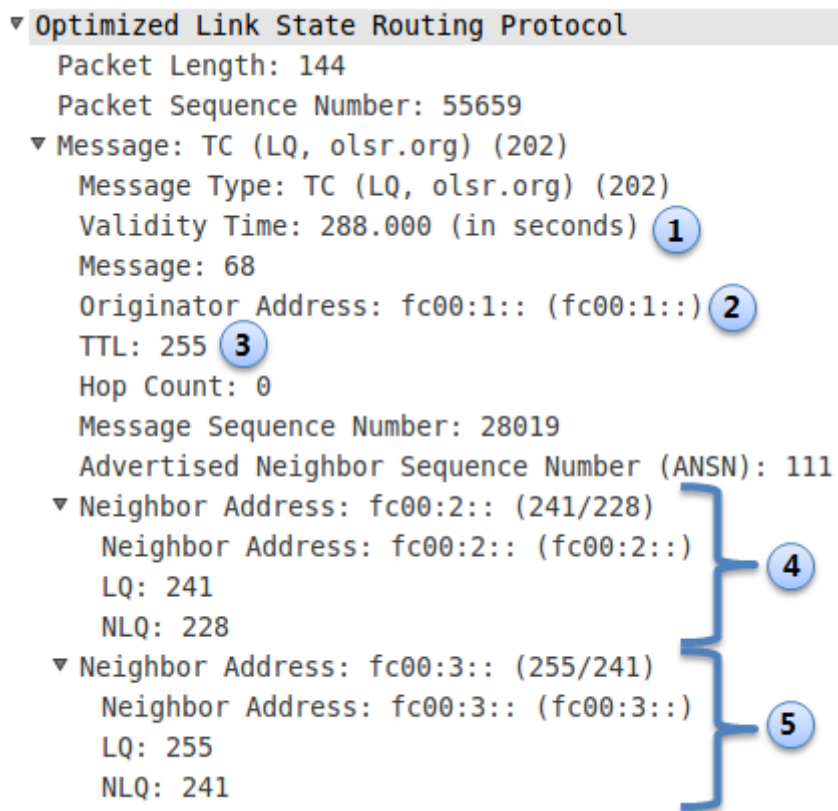


Obrázek č. 8.2: Analýza HELLO zprávy

1. Validity time – čas platnosti HELLO zprávy.
2. Originator Address – zdrojová adresa odesílatele HELLO zprávy.
3. TTL (Time To Live) – doba přežití paketu v síti, pokud je hodnota TTL nastavena na 1, pak je platnost paketu je jen v rámci sousedního uzlu, zpráva není přeposílána nikam dále.
4. Hello Emission Interval – interval zasílání HELLO paketů, nastaven na 2 vteřiny.
5. Neighbor Address – adresa sousedního uzlu - fc00:2::
6. Neighbor Address – adresa sousedního uzlu - fc00:3::

### 8.1.2 Analýza TC zprávy

Na obrázku č. 8.3 je znázorněn OLSR paket obsahující TC zprávu.

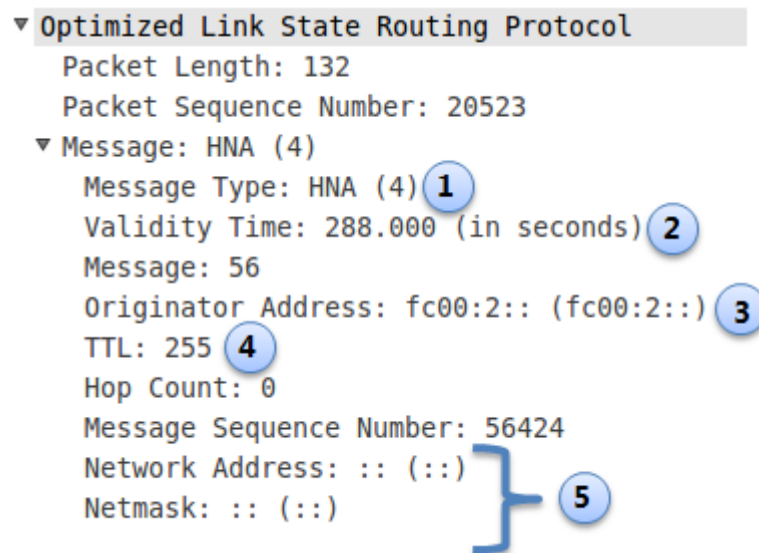


Obrázek č. 8.3: Analýza TC zprávy

1. Validity time – čas platnosti TC zprávy, nastaven na 288 vteřin.
2. Originator Address – zdrojová adresa odesílatele TC zprávy.
3. TTL (Time To Live) – doba přežití paketu v síti, pokud je hodnota TTL nastavena na 255, znamená to, že je paket možno přeposílat dále (255 přeskoků).
4. Neighbor Address – adresa sousedního uzlu - fc00:2::
5. Neighbor Address – adresa sousedního uzlu - fc00:3::

### 8.1.3 Analýza HNA zprávy

Význam HNA zprávy a jakým uzlem je zasílána je popsáno v kapitole 7.2.6. Na obrázku č. 8.4 je zachycena HNA zpráva. Zpráva HNA může být zasílána společně s HELLO zprávou.



Obrázek č. 8.4: Analýza HNA zprávy

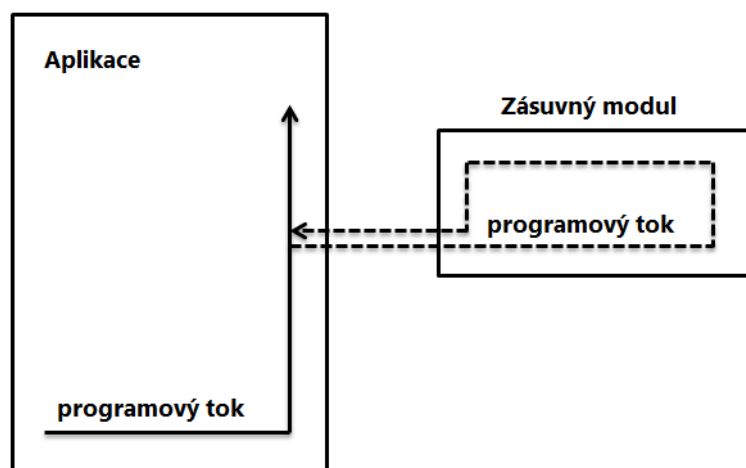
1. Message Type – typ OLSR zprávy (HNA - 4)
2. Validity time – čas platnosti HNA zprávy, nastaven na 288 vteřin
3. Originator Address – zdrojová adresa odesílatele HNA zprávy
4. TTL (Time To Live) – doba přežití paketu v síti, pokud je hodnota TTL nastavena na 255, znamená to, že je paket možno přeposílat dále (255 přeskoků)
5. Network Address a Netmask – adresa sítě a maska sítě, zápis a význam „::“ je vysvětlen v kapitole 1.2.6.



## 9 ZÁSUVNÉ MODULY OLSR DÉMONA

### 9.1 Princip zásuvných modulů

Pomocí zásuvných modulů je možno rozšířit funkční vlastnosti OLSR démona. OLSR démon podporuje tvorbu zásuvných modulů označovaných jako dynamicky linkované knihovny. Pomocí zásuvných modulů je možno definovat například novou sktrukturu OLSR paketu, nebo přidat funkce k původním, již definovaných paketům. Dynamicky linkované knihovny existují pro všechny běžné operační systémy, v Linuxu jsou označovány jako **.so** soubory. Na obrázku 9.1 je zobrazeno, jak zásuvný modul ovlivňuje aplikaci (např. OLSR démona).



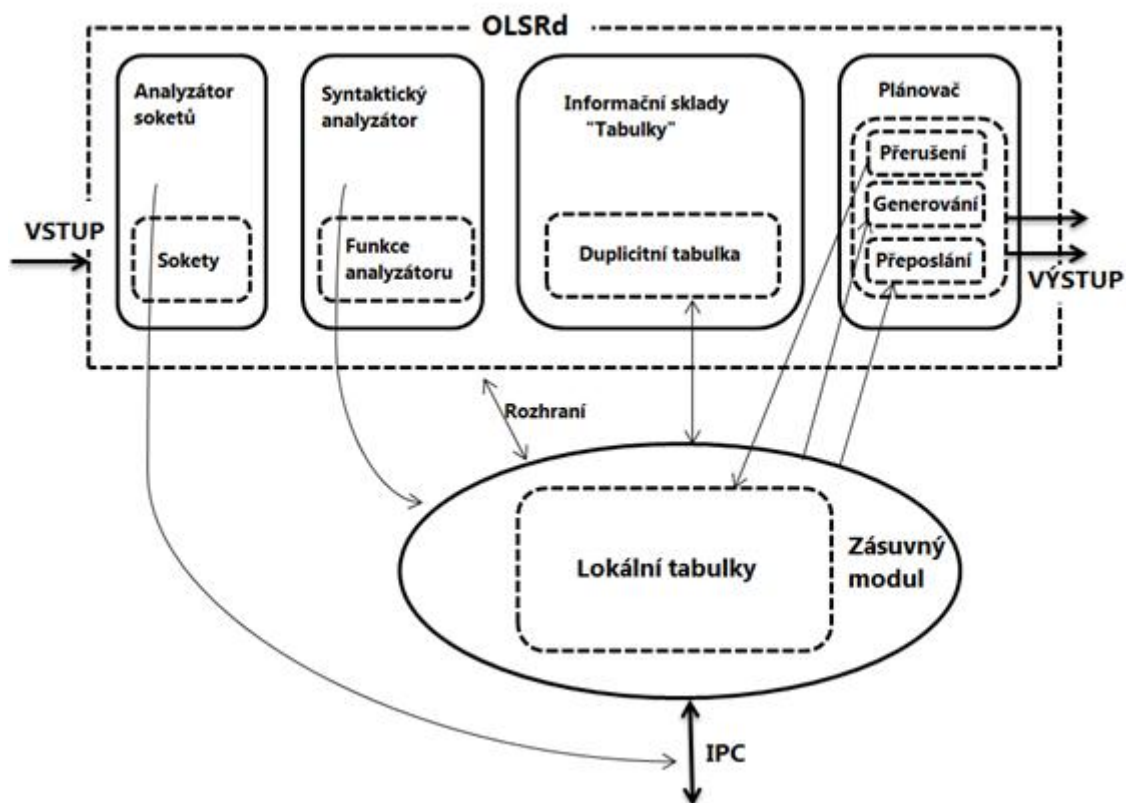
Obrázek 9.1: Interakce zásuvného modulu a aplikace

Hlavní výhody použití zásuvných modulů OLSR démona:

- a) Není nutno měnit kód OLSR démona, aby bylo možno přidat nový typ paketu
- b) Uživatelé mohou volně implementovat zásuvné moduly a licencovat si je tak, jak uznají za vhodné
- c) Zásuvné moduly mohou být napsány v jakémkoli programovacím jazyce, který může být poté zkompilován jako dynamicky linkovaná knihovna.

- d) Rozhraní zásuvných modulů je vždy kompatibilní, i když je vydána nová verze OLSR démona.

Na obrázku 9.2 je naznačeno, jak může zásuvný modul virtuálně komunikovat téměř se všemi částmi OLSR démona.



Obrázek č. 9.2: Rozhraní pro přístup zásuvného modulu

## 9.2 Návrh zásuvného modulu s podporou kvality služeb

Nový zásuvný modul se bude starat o rozesílání informací o zpoždění a volné šířce pásma mezi zařízeními pracujícími na protokolu OLSR. Informace o těchto parametrech budou rozesílány v OLSR paketech pomocí OLSR zpráv. Jeden OLSR paket může obsahovat více než jednu OLSR zprávu.

### 9.2.1 Návrh OLSR zprávy s podporou kvality služeb

Návrh OLSR zprávy nesoucí informace potřebné k zajištění požadované kvality služeb je zobrazen na obrázku č. 9.3. Význam jednotlivých polí bude dále vysvětlen.

	0b	8b	16b	32b
Hlavička OLSR paketu:	Délka paketu		Sekvenční číslo paketu	
OLSR zpráva:	Typ zprávy	Platnost	Délka zprávy	
	Tvůrce zprávy			
	TTL	Počet skoků	Sekvenční číslo zprávy	
	Cílová adresa			
	Zpoždění		Šířka pásma	

Obrázek č. 9.3: Zpráva OLSR obsahující nově definovaná pole

Význam jednotlivých polí je rozebrán v kapitole 2.4.4, nově definovaná pole jsou rozebrána níže.

Cílová adresa – adresa uzlu (IPv4 nebo IPv6), na obrázku zobrazena verze IPv4 (32b), na jehož lince mezi daným a cílovým uzlem je dané zpoždění. Nejedná se o adresu, kam budou informace zasílány, jelikož informace o stavu linky mezi těmito dvěma uzly musí znát celá síť.

Zpoždění – zpoždění uváděné v milisekundách na lince mezi daným uzlem a uzlem, jehož IP adresa je uvedena jako cílová adresa

Šířka pásma – volná šířka pásma daného uzlu uváděna v kilobajtech.

## 9.3 Rozbor zásuvného modulu s podporou kvality služeb

Zásuvný modul se skládá z několika souborů:

- a) `qos_messages.h` – obsahuje definici nové OLSR zprávy a obecného OLSR paketu
- b) `olsrd_plugin.c` – obsahuje funkce pro komunikaci s hlavním programem
- c) `olsrd_qos.h` – hlavičkový soubor obsahující inicializační funkce
- d) `olsrd_qos.c` – zde jsou definovány hlavní funkce pro plnění paketu, časovač atd.

V této části budou popsány pouze hlavní funkce obsažené v daných souborech, nebude zde prováděn žádný kompletní výpis kódu a rozbor jednotlivých položek. Soubor `qos_messages.h` obsahuje definované struktury, které reprezentují jednotlivé části OLSR zprávy. Soubor obsahuje dva typy struktur, jednu pro protokol IPv4 a druhou pro podporu protokolu IPv6.

### IPv6:

```
struct s_olsrmsg6 {
    uint8_t type;
    uint8_t olsr_vtime;
    uint16_t olsr_msgsize;
    struct in6_addr originator;
    uint8_t ttl;
    uint8_t hopcnt;
    uint16_t seqno;
    uint16_t zpozdeni;
    uint16_t sirka_pasma;
    struct in6_addr cislova_adresa;
};
```

### IPv4:

```
struct s_olsrmsg {
    uint8_t olsr_msgtype;
    uint8_t olsr_vtime;
    uint16_t olsr_msgsize;
    uint32_t originator;
    uint8_t ttl;
    uint8_t hopcnt;
    uint16_t seqno;
    uint16_t zpozdeni;
    uint16_t sirka_pasma;
    uint32_t cislova_adresa;
};
```

Struktura tedy obsahuje všechny povinné části OLSR zprávy a také společně s nimi nově definované položky ***zpozdeni***, ***sirka\_pasma*** a ***cislova\_adresa*** (IPv4 nebo IPv6). Tato struktura se nyní doplní do další struktury, která obsahuje hlavičku OLSR paketu. Nově definovaná zpráva je obsažena ve struktuře *s\_olsrmsg* nebo *s\_olsrmsg6* jako nová OLSR zpráva.

#### IPv6:

```
struct s_olsr6 {
    uint16_t olsr_packlen;
    uint16_t olsr_seqno;
    struct s_olsrmsg6 olsr_msg[1];
};
```

#### IPv4:

```
struct s_olsr {
    uint16_t olsr_packlen;
    uint16_t olsr_seqno;
    struct s_olsrmsg olsr_msg[1];
};
```

Soubor *olsrd\_plugin.c* tedy obsahuje funkce pro komunikaci s hlavním programem. Funkce, které se starají o tuto komunikaci, musí být obsaženy v každém zásuvném modulu a jedná se o tyto funkce:

**int** olsrd\_plugin\_init() - stará se o inicializaci zásuvného modulu

**int** olsrd\_plugin\_interface\_version() - vrací hodnotu, která definuje verzi protokolu

**void** my\_init() - konstruktor

**void** my\_fini() - destruktork

**int** store\_plugin\_int() - uchovává staticky definované parametry

**struct** olsrd\_plugin\_parameters plugin\_parameters[] - struktura obsahující staticky definované parametry

**void** olsrd\_get\_plugin\_parameters() - funkce pro získání přístupu ke staticky definovaným parametrům

Soubor *olsrd\_plugin.h*, jedná se o hlavičkový soubor, obsahuje tři inicializační funkce, jedná se o definice funkcí využívaných v souboru *olsrd\_plugin.c*.

```
int qos_plugin_init()
void qos_plugin_exit()
int plugin_ipc_init()
```

Nejdůležitějším souborem je soubor *olsrd\_qos.c*, který obsahuje hlavní funkce pro tvorbu paketu, zacházení s paketem, časování zpráv atd. Soubor obsahuje tyto funkce:

```
int qos_build() - tato funkce odkazuje na funkci build_qos6()

int parse_qospacket() - jedná se o funkci, která má na starost zpracování nově přijatého
                        paketu, funkce zjistí, jestli existuje daný záznam a pokud ne.
                        tak vytvoří v databázi novou položku

int build_qos6() - funkce, která naplňuje nový OLSR paket OLSR zprávou – IPv6

int build_qos() - funkce, která naplňuje nový OLSR paket OLSR zprávou – IPv4

int get_delay() - funkce vrací hodnotu zpoždění – není plně implementována

int get_bandwidth() - funkce vrací hodnotu volné šířky pásma - není plně implementována

struct qos_stamp *lookup_timestamp_entry()- tato funkce kontroluje a spravuje
                        záznamy QoS parametrů v databázi

void generate_qos() - tato funkce generuje nový OLSR paket

olsr_start_timer() - jedná se o časovač, který spouští periodické generování OLSR paketů
```

## 9.4 Instalace a konfigurace QoS zásuvného modulu

Aby bylo možno nový zásuvný modul využít je nutno tento modul nejprve zkompileovat a nainstalovat. Modul ovšem není možno kompilovat kdekoli ale je ho nutno umístit do složky **/lib** v adresáři OLSRd. Celá cesta umístění adresáře **/lib** je **/home/olsr/olsrd/**. Pokud je zásuvný modul správně umístěn, tak pro správné zkompileování a pro správnou instalaci je potřeba zadat následující příkazy do příkazové řádky terminálu:

```
# cd /home/olsr/olsrd/lib/zvolený_název_složky (např. qos_plugin)
```

```
# sudo make install
```

Konfigurace nového zásuvného modulu se provádí v konfiguračním souboru OLSR démona.

```
# sudo nano /etc/olsrd.conf
```

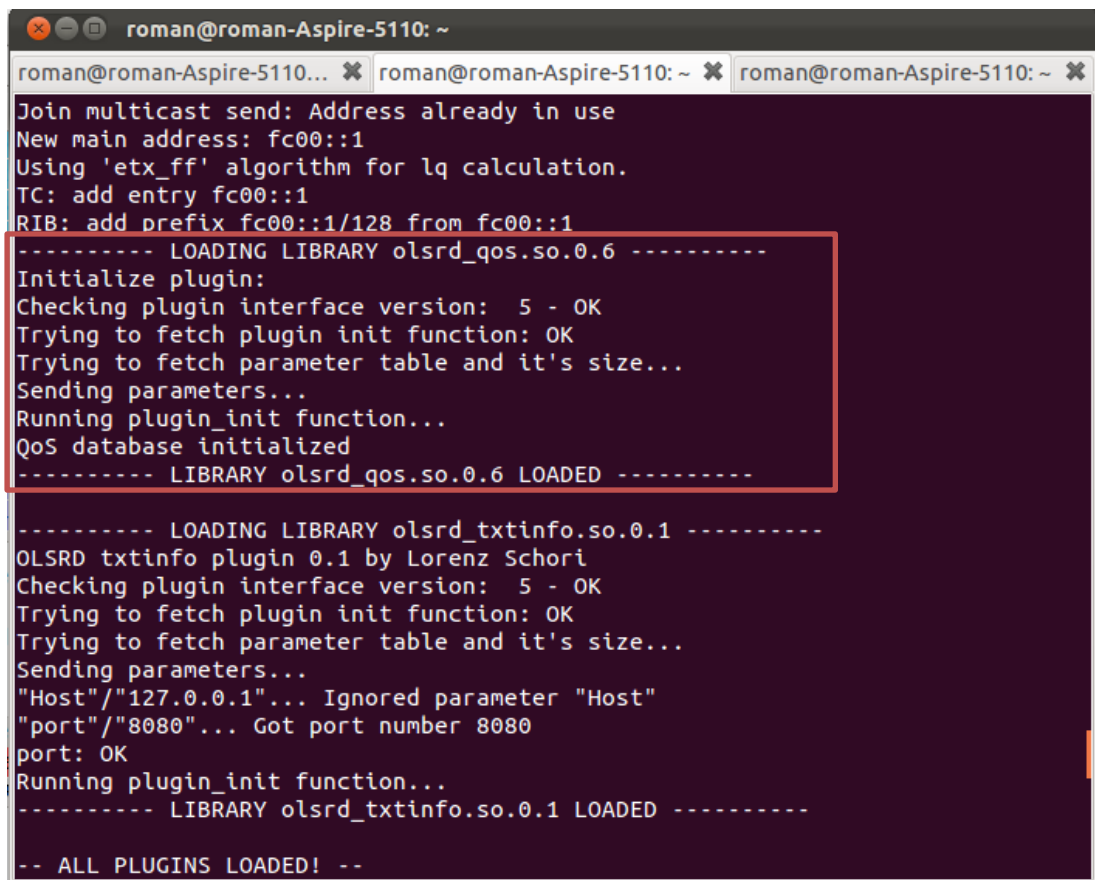
V konfiguračním souboru je potřeba dopsat v sekci, která se stará o načítání zásuvných modulů, viz kapitola 7.2.2 následující řádek:

```
LoadPlugin "olsrd_qos.so.0.6" { }
```

Nyní je možno nový zásuvný modul plně využívat. Pro ověření správného načtení nově vytvořeného zásuvného modulu je potřeba spustit OLSR démona příkazem:

```
# sudo olsrd
```

Na obrázku 9.4, je pomocí výpisu konzole zobrazeno korektní inicializace nového zásuvného modulu.



```
roman@roman-Aspire-5110: ~  
roman@roman-Aspire-5110... ✕ roman@roman-Aspire-5110: ~ ✕ roman@roman-Aspire-5110: ~ ✕  
Join multicast send: Address already in use  
New main address: fc00::1  
Using 'etx_ff' algorithm for lq calculation.  
TC: add entry fc00::1  
RIB: add prefix fc00::1/128 from fc00::1  
----- LOADING LIBRARY olsrd_qos.so.0.6 -----  
Initialize plugin:  
Checking plugin interface version: 5 - OK  
Trying to fetch plugin init function: OK  
Trying to fetch parameter table and it's size...  
Sending parameters...  
Running plugin_init function...  
QoS database initialized  
----- LIBRARY olsrd_qos.so.0.6 LOADED -----  
  
----- LOADING LIBRARY olsrd_txtinfo.so.0.1 -----  
OLSRD txtinfo plugin 0.1 by Lorenz Schori  
Checking plugin interface version: 5 - OK  
Trying to fetch plugin init function: OK  
Trying to fetch parameter table and it's size...  
Sending parameters...  
"Host"/"127.0.0.1"... Ignored parameter "Host"  
"port"/"8080"... Got port number 8080  
port: OK  
Running plugin_init function...  
----- LIBRARY olsrd_txtinfo.so.0.1 LOADED -----  
  
-- ALL PLUGINS LOADED! --
```

Obrázek č. 9.4: Výpis konzole



Na následujícím obrázku je znázorněn výpis konzole při korektním spuštění nového zásuvného modulu. Zobrazeny jsou jednotlivé kroky, které jsou během programu spuštěny, tedy například hledání síťových rozhraní, naplnění paketu atd.

```
roman@roman-Aspire-5110: ~
*** olsr.org - pre-0.6.3-git_0081402-hash_36723d6e66f8ac59d49044c0721352
75 - (2012-02-28 18:52:31 on roman-Aspire-5110) ***

--- 17:05:04.314202 ----- LINKS

IP address      hyst      LQ      ETX

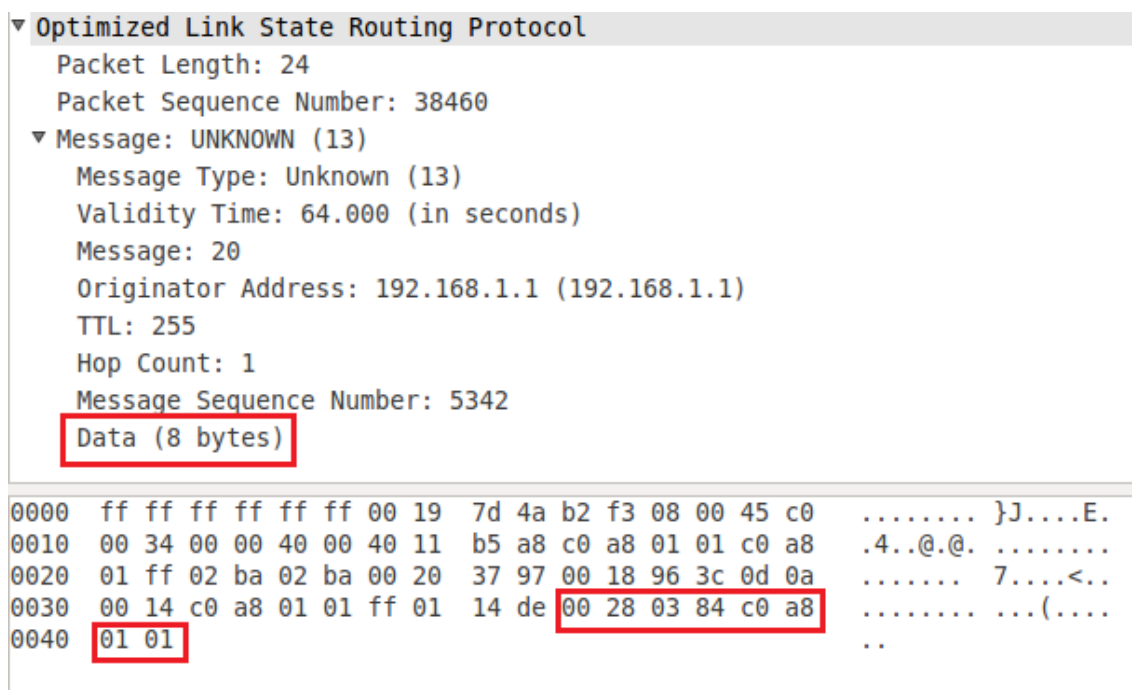
--- 17:05:04.314331 ----- TWO-HOP NEIGHBORS

IP addr (2-hop)  IP addr (1-hop)  Total cost
Hledám síťová rozhraní
Sekvenční číslo 45867
Hodnota zpoždění 30 ms
Hodnota šířky pásma 800 Kbs
Velikost OLSR zprávy: 20
QoS zpráva naplněna
Paket odeslán
-
```

Obrázek č. 9.5: Výpis konzole po spuštění hlavního programu

## 9.5 Analýza OLSR paketu s obsahem nové OLSR zprávy

Nově vytvořenou OLSR zprávu je možno zachytit například pomocí programu Wireshark. Zachycení OLSR paketu s novou OLSR zprávou je zobrazeno na obrázku 9.6. Zachycený paket bohužel neobsahuje položky cílová adresa, zpoždění a volná šířka pásma, je to z důvodu, že program Wireshark nemá tyto nové položky nadefinovány a tedy je nemůže korektně zobrazit. Nově definovaná pole jsou ukryta pod označením Data (8B). Těchto 8B je tedy dáno součtem cílové IPv4 adresy (4B), zpoždění (2B) a volné šířky pásma (2B). Z tzv. raw paketu je možno vyčíst tyto nově definované položky, tyto položky jsou uváděny v hexadecimálním tvaru.



Obrázek 9.6 : Struktura nové OLSR zprávy

## ZÁVĚR

Téměř každé přenosné zařízení je v dnešní době schopno komunikovat s ostatními přenosnými i nepřenosnými zařízeními pomocí bezdrátové technologie. Díky tomuto předpokladu, je potenciál využití MANET sítí velmi velký. MANET síť si můžeme představit jako dynamicky vytvářenou bezdrátovou síť, která se vytváří ze skupiny mobilních uzlů, které mezi sebou komunikují na sdíleném bezdrátovém komunikačním kanále. Zajištění požadované kvality služeb je v tomto typu sítě velice obtížným úkolem. Mezi hlavní problémy se zajištěním kvality služeb můžeme zařadit například složitou údržbu komunikační cesty, postrádání centralizované kontroly, mobilitu uzlů nebo omezenou kapacitu energie komunikačních uzlů. Existují metody, jak požadovanou kvalitu služeb v MANET sítích zaručit. V této práci byly rozebrány metody zajištění kvality služeb pomocí směrovacího protokolu OLSR s podporou kvality služeb a signalizačního protokolu INSIGNIA.

Praktická část práce se zaměřuje na otestování podpory kvality služeb v MANET sítích. Pomocí programu OPNET Modeler byly vytvořeny dva scénáře simulace MANET sítě, první bez podpory kvality služeb a druhý s podporou kvality služeb. Výsledky simulací byly zaměřeny na rozdíly v zajištění požadavků na kritické parametry, jako jsou kolísání zpoždění hlasového přenosu, kvalita hlasového přenosu, zpoždění hlasového přenosu a zpoždění video přenosu. Výsledky simulací byly přehledně graficky zpracovány. Z výsledků simulací je zřetelné, že podpora kvality služeb má v MANET sítích velký význam a přispívá k lepším vlastnostem hlasového i video přenosu.

Nasazení protokolu OLSR do reálné MANET sítě bylo zajištěno pomocí implementace UniK OLSR. Pomocí programu Wireshark byl zachycen provoz protokolu OLSR, jednotlivé zprávy zasílané v rámci tohoto protokolu byly detailně rozebrány. Posledním úkolem bylo prostudovat implementaci protokolu OLSR a navrhnout určité modifikace. Tento cíl byl splněn pomocí návrhu nového zásuvného modulu, který se stará o zasílání informací o stavu linky mezi jednotlivými uzly MANET sítě. Zprávy zasílané tímto zásuvným modulem byly úspěšně zachyceny a analyzovány pomocí programu Wireshark.

## LITERATURA

- [1] A. Munaretto, H. Badis, K. Al Agha and G. Pujolle, "A Link-state QoS Routing Protocol for Ad hoc Networks," In the proceedings of IEEE MWCN2002, Stockholm, Sweden, September 2002.
- [2] AODV-UU. *Sourceforge* [online]. © 2012 [cit. 2012-02-17]. Dostupné z: <<http://sourceforge.net/projects/aodvu/>>
- [3] Babel — a loop-avoiding distance-vector routing protocol. *Laboratoire Preuves, Programmes et Systèmes* [online]. [cit. 2012-02-17]. Dostupné z: <<http://www.pps.jussieu.fr/~jch/software/babel/>>
- [4] BADIS, H., KHALDOUN, A. Quality of Service for Ad hoc Optimized Link State Routing Protocol (QOLSR) [online]. Institut Gaspard-Monge, Marne-la-Vallée, France, March 2007. 37s. Dostupné z URL: <<http://ietfreport.isoc.org/cgi-bin/id2pdf?f1=draft-badis-manet-qolsr05.txt>>.
- [5] CLAUSEN, T., JACQUET, P. Optimized Link State Routing Protocol (OLSR) [online]. 2003, RFC 3262, IETF Network Working Group. Dostupné z URL: <<http://portal.acm.org/citation.cfm?id=RFC3626>>.
- [6] Distribuce Ubuntu 11.10: < <http://www.ubuntu.cz/ziskejte/stahnout>>
- [7] FENG, Qian, Zhongmin CAI, Jin YANG a Xunchao HU. A Performance Comparison of the Ad Hoc Network Protocols. In: *A Performance Comparison of the Ad Hoc Network Protocols* [online]. 28.10 2009 [cit. 2012-02-17]. Dostupné z: <[http://ieeexplore.ieee.org/xpl/freeabs\\_all.jsp?arnumber=5403292](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5403292)>
- [8] Hannan XIAO, Winston K.G. SEAH, Anthony LO and Kee ChiangCHUA "Flexible Quality Service Model for Ad-HOC Networks". Inproceedings of the IEEE Vehicular Technology Conference, Tokio, Japan, May 2000.
- [9] IETF RFC: 1633, Integrated Services in the Internet Architecture: an Overview.
- [10] ILYAS, Mohammad . *The Handbook Of Ad-hoc Wireless Networks* [online]. Boca Raton, Florida: CRC PRESS, 2003 [cit. 2011-10-27]. Dostupné z WWW: <<http://www.scribd.com/doc/54148816/adhoc>>.

- [11] Implementation of Dynamic Source Routing. *Monarch Project: Mobile Networking Architectures*[online]. 2000 [cit. 2012-02-17]. Dostupné z: <<http://www.monarch.cs.rice.edu/dsr-impl.html>>
- [12] Lee, S. and A. Campbell, *INSIGNIA: In-band Signaling Support for QOS in Mobile Ad Hoc Networks*, Proc of 5th International Workshop on Mobile Multimedia Communications (MoMuC,98), Berlin, Germany, October 1998.
- [13] M. Frodigh, P. Johansson, and P. Larsson. "Wireless ad hoc networking: the art of networking without a network," Ericsson Review, No. 4, 2000, pp. 248-263.
- [14] MOLNÁR, Karol. ZEMAN, O., SKOŘEPA, M., Moderní síťové technologie, Laboratorní cvičení [online]. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2008. 101 s.  
Dostupné z URL: <[http://www.utko.feec.vutbr.cz/~molnar/mmms/MMOS\\_lab.pdf](http://www.utko.feec.vutbr.cz/~molnar/mmms/MMOS_lab.pdf)>.
- [15] NOVOTNÝ, V. Mobilní směrovací protokoly s podporou IPV6 (MANET) [online]. 2007, Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2007. 89 s. Vedoucí práce byl Ing. Michal Soumar.  
Dostupné z URL: <<http://docs.google.com/viewer?a=v&pid=wave&srcid=8QI35XxE1&chrome=true&pli=1>>.
- [16] OPNET Technologies, OPNET Modeler Product Documentation Release 16.0, OPNET Technologies Inc., 2010.
- [17] PUŽMANOVÁ, R. 2005. *Schválena specifikace pro hlas do WiFi*. [Online] 2005. Dostupný z: <<http://www.lupa.cz/clanky/schvalena-specifikace-pro-hlasdo-wifi>>.
- [18] QOLSR: QoS with the OLSR protocol. *QOLSR* [online]. [cit. 2012-02-17]. Dostupné z: <<http://qolsr.lri.fr/>>
- [19] RFC 5498. IANA Allocations for Mobile Ad Hoc Network (MANET) Protocols <<http://tools.ietf.org/html/rfc5498>>

- [20] S.-B. Lee et al.. "INSIGNIA: An IP-Based Quality of Service Framework for Mobile Ad-Hoc Networks." *1. Parallel and Dist. Comp* , Special issue on Wireless and Mobile Computing and Communications, vol.60 no 4, Apr 2000, p. 374-406.
  
- [21] SRIDHAR, S.; BASKARAN, R. A Survey on QoS Based Routing Protocols for MANET *International Journal of Computer Applications* [online]. , October 2010, Volume 8– No.3, [cit. 2011-10-27]. Dostupný z WWW: <<http://www.ijcaonline.org/volume8/number3/pxc3871671.pdf>>.
  
- [22] SANTHI, G.; NACHAPPAN, Alamelu. A Survey Of QoS Routing Protocols For Mobile Ad-hoc Networks. *International Journal of Computer Science and Information Technology*. August 2010, Vol.2, No. 4, s. 125-132
  
- [23] [Www.olsr.org](http://www.olsr.org). *Olsrd an adhoc wireless mesh routing daemon* [online]. © 2004-2008 [cit. 2012-02-17]. Dostupné z: <<http://www.olsr.org>>

## **SEZNAM ZKRATEK**

AODV - Ad hoc On-Demand Distance Vector Routing

AQOR - Ad hoc QoS on-demand routing

BAN - Body Area Network

BE - Best Effort

BL - Basic Layer

CEDAR - Core Extraction Distributed Ad hoc Routing

DARPA - Defence Advanced Research Project Agency

DBMS - Database Management System

DSR - Dynamic Source Routing

EL - Enhanced Layer

FTP – File Transfer Protocol

HCF - Hybrid Coordination Function

HNA - Host Network Association

IPv4 - Internet Protocol version 4

IPv6 - Internet Protocol version 6

LAN - Local Area Network

LMH Large Mobile Host

MANET - Mobile Ad-hoc Network

MID - Multiple Interface Declaration

MOS - Mean Opinion Score

MPR – Multipoint Relaying

NS 2 - Network Simulator 2

OLSR - Optimized Link State Routing protocol

PAN - Personal Area Network

PHB - Per-Hop-Behaviour

PLBQR - Predictive Location-based QoS Routing

QAODV - QoS Ad-Hoc On-Demand Distance Vector

QMPR - QoS - MPR

QMRPD - QoS Multicast Routing Protocol for Dynamic group topology

QOLSR - QoS Optimized Link State Routing protocol

QoS – Quality of Services

RSVP - Resource Reservation Protocol

RT - Real Time

SMH - Small Mobile Host

TBR - Tree-Based Routing

TORA - Temporally-Ordered Routing Algorithm

VoIP – Voice over IP

WAN - Wide Area Network

WLAN - Wireless Local Area Network

ZRP - Zone Routing Protocol